

2

THE BRADLEY DEPARTMENT OF ELECTRICAL ENGINEERING

VIRGINIA TECH

AD-A251 910



DTIC
ELECTE
JUN 18 1992
S C D

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited



92-14997



92 6 08 014

VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY
Blacksburg, Virginia 24061 (703)231-6646

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

| | | | | |
|---|---|--|--|--|
| 1. AGENCY USE ONLY (Leave blank) | | 2. REPORT DATE May 15, 1992 | 3. REPORT TYPE AND DATES COVERED unclassified 1989-1991 | |
| 4. TITLE AND SUBTITLE Final Report ONR Grant N00014-89-J-3123 Active Control of Generalized Complex Modal Structures in a Stochastic Environment | | | 5. FUNDING NUMBERS N00014-89-J-3123 | |
| 6. AUTHOR(S) Richard L. Moose William T. Baumann Hugh F. VanLandingham | | | | |
| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Department of Electrical Engineering VPI&SU Blacksburg, VA 24061-0111 | | | 8. PERFORMING ORGANIZATION REPORT NUMBER | |
| 9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Applied Research and Technology Directorate Office of Naval Research 800 N. Quincy Street Arlington, VA 22217-5000 | | | 10. SPONSORING/MONITORING AGENCY REPORT NUMBER | |
| 11. SUPPLEMENTARY NOTES | | | | |
| 12a. DISTRIBUTION/AVAILABILITY STATEMENT See Attachment | | | 12b. DISTRIBUTION CODE | |
| 13. ABSTRACT (Maximum 200 words) This report deals with the active control of heavily damped vibratory mechanical systems in a stochastic environment. The systems of interest are multiple input, multiple output (MIMO) structures that are modeled by generalized complex modes. The control objective is to suppress the effects of impulsive disturbances within a short period of time as well as to provide specified reduction of vibration due to ever-present Gaussian and nonGaussian stochastic disturbances. The team effort, described separately, in this report can be viewed as three interactive parts: one group investigating the robust control issues; one, the non-gaussian estimation problem and a third group working with a mechanical plate experiment, generating data and identification algorithms. The experimental model was constructed in the M. E. Department under DARPA funding. The robust control investigators used modern analysis and synthesis tools to incorporate the required performance directly into the design procedure. The approach to handle the non-gaussian state estimation problem was in the use of an adaptive Gaussian sum state estimator. | | | | |
| 14. SUBJECT TERMS | | | 15. NUMBER OF PAGES 374 | |
| | | | 16. PRICE CODE | |
| 17. SECURITY CLASSIFICATION OF REPORT unclassified | 18. SECURITY CLASSIFICATION OF THIS PAGE unclassified | 19. SECURITY CLASSIFICATION OF ABSTRACT unclassified | 20. LIMITATION OF ABSTRACT none | |

ATTACHMENT
REPORTS DISTRIBUTION

ADDRESSEES

NUMBER OF COPIES

Scientific Officer Code: 1125GG
Joseph H. Kravitz
Office of Naval Research
800 North Quincy Street
Arlington, Virginia 22217-5000

3

Administrative Grants Officer
Office of Naval Research
Resident Representative N68883
Charles S. Draper Laboratory
555 Technology Square MS54
Cambridge, MA 02139-3539

1

Director, Naval Research Laboratory
ATTN: Code 2627
Washington, DC 20375

1

Defense Technical Information Center
Building 5, Cameron Station
Alexandria, Virginia 22304-6145

2



Statement A per telecon Dr. Eric Hendricks
ONR/Code 1221
Arlington, VA 22217-5000

NWW 6/17/92

| | |
|--------------------|--|
| Accession For | |
| NTIS GRA&I | <input checked="checked" type="checkbox"/> |
| DTIC TAB | <input type="checkbox"/> |
| Unannounced | <input type="checkbox"/> |
| Justification | |
| By _____ | |
| Distribution/ | |
| Availability Codes | |
| Dist | Avail and/or Special |
| A-1 | |

FINAL REPORT

May 15, 1992

ONR GRANT NO. N00014-89-J-3123

**Active Control of Generalized Complex Modal Structures
in a Stochastic Environment**

Investigators:

William T. Baumann (Co-P.I.)
Richard L. Moose (Co-P.I.)
Hugh F. VanLandingham (Co-P.I.)
Mauro J. Caputi (Ph.D. Student)
Stephen H. Jones (Ph.D. Student)
Bhaskar Gorti (M.S. Student)

The Bradley Department of Electrical Engineering
Virginia Polytechnic Institute and State University
Blacksburg, Virginia 24061-0111

Program Manager:

Dr. Eric Hendricks
Applied Research and Technology Directorate
Office of Naval Research
Applied Research Division
800 N. Quincy Street
Arlington, Virginia 22217-5000

CONTENTS

| | |
|------------------------------------|----------|
| Summary | 1 |
| 1. Technical Objectives | 1 |
| 2. Technical Approaches | 1 |
| 3. Summary of Accomplishments | 1 |
| 4. Significance of Accomplishments | 4 |
| 5. Publications and Presentations | 4 |
| 6. Participant List | 5 |

Appendix A

"Parameter Robust Reduced-Order Control of Flexible Structures", Ph. D. Dissertation, S. H. Jones.

Appendix B

"Non-Gaussian Estimation Using a Modified Gaussian-Sum Adaptive Filter", Ph. D. Dissertation, M. J. Caputi.

Appendix C

[Copies of Publications Supported by this Contract]

SUMMARY

(1) TECHNICAL OBJECTIVES AND ISSUES UNDER INVESTIGATION:

Working with realistic modal descriptions of vibratory mechanical systems, the technical objective was to design active damping controllers. The systems of interest are multiple input, multiple output (MIMO), heavily damped structures with complex modes. The control objective is to suppress the effects of impulsive disturbances within a short period of time as well as to provide a specified reduction of vibration due to ever-present stochastic disturbances.

Systems with complex modes cannot be decoupled exactly. One issue that was resolved was the impracticality of using a controller for an approximately decoupled system. Simulations showed that the desired amount of damping could not be achieved with this controller due to stability problems caused by the approximations.

The technical issue regarding state estimation was to develop an optimal estimator for the system driven by non-gaussian stochastic signals.

(2) TECHNICAL APPROACHES:

The team effort can be viewed as three interactive parts: one group investigating the robust control issues; one, the non-gaussian estimation problem and a third group working with a mechanical plate experiment, generating data and identification algorithms. The experimental model was constructed in the M.E. Department under DARPA funding. Our group interacted well with the M.E. group, sharing ideas and methods where relevant.

The robust control investigators used modern analysis and synthesis tools to incorporate the required performance directly into the design procedure. The approach to handle the non-gaussian estimation problem was the use of a modified gaussian sum.

(3) SUMMARY OF ACCOMPLISHMENTS:

General Discussion

Our effort began with the design of a baseline controller. The system of interest was a MIMO, heavily damped structure with complex modes, and the control objective was to suppress impulsive disturbances in a given time interval and to provide a specified degree of suppression to an ever-present stochastic disturbance for which we have frequency domain information. In order to limit the complexity of our controller, a decoupled, independent modal control approach was explored. Although a system with complex modes cannot be decoupled exactly, an attempt

was made to design a controller for an approximately decoupled system. Simulations showed, however, that the desired increase in damping could not be achieved with this controller due to stability problems brought about by the approximations. The decoupling approach was abandoned and replaced by a standard estimator-feedback control structure. To address both types of disturbances simultaneously, an LQG/LTR methodology was used, after suitable modification to account for the presence of the feed-through term in our system that was due to the use of accelerometers as sensors. This provided an acceptable baseline solution to our problem to which we could compare our ideas for improvement. One area in which the baseline design was deficient was robust stability to unstructured uncertainty. The solution to this problem was to incorporate robust stability and performance requirements directly into the design procedure.

The Problem of Structured Uncertainties

Robust control methods were studied for application to vibrational systems whose uncertainties have known structure. Structured uncertainties arise, for example, in plants with uncertain parameters in the state space model. Therefore, vibrational systems with uncertain modal frequencies and damping ratios are well suited for structured uncertainty designs. The purpose of such methods is to take advantage of the structure of plant uncertainties in order to reduce the conservativeness of the compensator design. Structured uncertainty designs accomplish this by guaranteeing stability robustness only to uncertainties of the specified structure, thereby sacrificing as little performance as possible.

The chosen design method is a refinement of a modified Linear Quadratic Gaussian method by Jong-Yin Lin and others. Lin models parameter uncertainties by isolating constants of unknown magnitude in fictitious feedback paths and treating the return signals as gaussian white noise entering the plant with direction determined by the nature of the uncertainty. The proposed method exploits the fact that these return signals are neither gaussian, nor white noise. Rather, they depend on the magnitudes of the states, controls, and measurements of the system. A more accurate modeling of the parameter uncertainties will normally yield compensator designs with better performance for the same level of stability robustness. This improvement is achieved without an increase in compensator order or a loss of optimality. Also, the optimal reduced-order compensator equations of Hyland and Bernstein adapt naturally to the proposed design method and allow an additional tradeoff between control system performance and compensator order reduction.

Results in Non-Gaussian Estimation

Optimal state estimation has been accomplished for a linear system driven by an unknown non-gaussian input with additive white noise, and observed by measurements containing feedthrough of the same nonGaussian input and corrupted by additional white Gaussian noise. One approach considered that can cope with the non-gaussian nature of the input signal is the "gaussian sum technique", where the probability density function of the non-gaussian input is approximated by a weighted sum of gaussian density functions. An adaptive filter based on this

technique, however, has the limitation that the number of terms in the gaussian sum grows at each iteration of the filter. A modified approach has been developed in this research which results in an adaptive filter with a fixed number of terms at each iteration.

As a result of this modified gaussian sum research, a necessary condition for effective estimation has been established. It has been found that the value of the DC gain of the linear system provides a simple test to determine whether or not the modified estimation scheme will prove effective. Several alternate estimation methods have been established when this condition is not met. Also, a suboptimal method of tuning the parameters of the adaptive filter structure to provide enhanced performance has been investigated.

This modified estimation technique proved effective in the problem of robust modal control of large, MIMO, heavily damped structures. The adaptive filter was found to provide high quality estimates of modal position and velocity required by the robust control system.

Progress on System Identification

Effective and efficient system identification techniques for discrete-time, linear, MIMO, heavily damped modal systems from input/output sequences have been developed and simulated. This facilitated a better understanding of the possible errors in the estimated model and a more accurate compensator and estimator design.

Three different time domain identification techniques have been developed in this research. The first technique is known as Pseudo-Linear Identification (PLID) for simultaneous state estimation and parameter identification. The method involves forming an augmented system that is nonlinear, due to the multiplication of the state by parameters. The augmented state vector is estimated by a Kalman filter, which has a state matrix that depends on the input and output data.

The second algorithm developed determines the state-space model in a pseudo-controllable/observable canonical form. The advantage of this technique is its ability to determine structural information, which means determination of controllability/observability indices. A MIMO system is obtained in pseudo controllable/observable form, based on a set of admissible pseudo- controllable/observable indices.

The last identification algorithm developed in this research is a computational simplification of the Eigensystem Realization Algorithm. The ERA algorithm involves singular value decomposition of the Hankel matrix formed by the Markov parameters. This Hankel matrix grows as the system order and number of inputs and outputs increase. With the computational simplification, a minimal state space representation is obtained by a simple and appropriate selection of the columns or rows from the Hankel matrix. The identified state space model is obtained in a canonical form.

These system identification techniques have been verified on a simply supported

rectangular plate experimental set-up. The results achieved were very encouraging. These techniques could be easily applied to a more complex modal structure.

(4) SIGNIFICANCE OF ACCOMPLISHMENTS

This project has provided results in the direction of a complete control algorithm for structural vibration damping. When the various research results came together, the control algorithm was found to be capable of reducing critical modes of vibration in response to persistent or impulsive non-gaussian disturbances.

(5) PUBLICATIONS AND PRESENTATIONS:

Copies of the following papers are included with this report.

Journal Papers:

"A Computationally Efficient Technique for State Estimation of Nonlinear Systems", Automatica, February 1992, (J.S.Dhingra, R.L.Moose, H.F.VanLandingham and T.A.Lauzon).

"Nonlinear Effects of a Modal Domain Optical Fiber Sensor in a Vibration Suppression Control Loop for a Flexible Structure", ASME Journal of Vibration and Acoustics, to appear. (D. K. Lindner, G. A. Zvonar, W. T. Baumann and P. L. Delos).

"A Modified Gaussian-Sum Approach to Estimation of Non-Gaussian Stochastic Signals", IEEE Transactions on Aerospace and Electronic Systems, to appear (July 1992). (M. J. Caputi and R. L. Moose).

International Conference:

"Measurement and Control of Flexible Structures Using Distributed Sensors", 29th IEEE Conference on Decision and Control, December 1990. (D.K.Lindner, K.M. Reichard, W.T.Baumann and M. F. Barsky).

"Application of Fuzzy Logic Control to Active Vibration Damping", Eighth VPI & SU Symposium on Dynamics and Control of Large Structures, May 1991. (A. Tsoukkas and H. F. VanLandingham).

National and Regional Conferences:

"Deterministic Identification of Linear Multivariable Systems", 22nd Southeastern Symposium on System Theory, March 1990. (B.Gorti, S.Bingulac and H.F.VanLandingham).

"A Computational Procedure for Stabilizing Lightly Damped Systems", 21st Modeling and Simulation Conference, May 1990. (B.Gorti, S.Bingulac and H.F.VanLandingham).

"Estimation of Non-Gaussian Signals Using a Modified Gaussian Sum Adaptive Filter", 28th Allerton Conference, October 1990. (M.J.Caputi, R.L.Moose and W.T.Baumann).

"Computational Simplification in Eigensystem Realization Algorithm", 28th Allerton Conference, October 1990. (B.Gorti, S.Bingulac and H.F.VanLandingham).

(6) PARTICIPANT LIST:

William T. Baumann, co-investigator

Richard L. Moose, co-investigator

Hugh F. VanLandingham, co-investigator

Stanoje Bingulac, consultant

Mauro J. Caputi, Ph.D. student, degree awarded in May 1991

Stephen H. Jones, Ph.D. student, degree awarded in July 1991

Bhaskar M. Gorti, M.S. student, degree awarded in February 1991

Table of Contents

| | Page |
|--|------|
| 0. Introduction | 1 |
| 0.1 Purpose and Importance of This Work | 1 |
| 0.2 Research Objectives | 2 |
| 1. Motivation for This Study | 4 |
| 1.1 Criteria for the Merit of Candidate Designs | 4 |
| 1.1A. Nominal Performance | 4 |
| 1.1B. Performance/Stability Robustness Tradeoff | 8 |
| 1.1C. Compensator Order | 16 |
| 1.1D. Steady State, Transient Disturbance Rejection | 17 |
| 1.2 Contributions | 17 |
| 2. Frequency-Shaped Noise..... | 19 |
| 2.1 Problem Statement | 19 |
| 2.2 Noise Shaping Filter Design..... | 21 |
| 2.3 Controller Design Summary | 23 |
| 3. Multiplicative White Noise..... | 25 |
| 3.1 Problem Statement | 25 |
| 3.2 Preliminaries of Stochastic Differential Equation Theory | 28 |
| 3.3 Conversion to Deterministic Minimization Problem..... | 30 |
| 3.4 Derivation of the Necessary Conditions | 35 |
| 3.5 Modification for Stratonovich Noise Model | 41 |
| 3.6 Controller Design Summary | 42 |
| 4. Optimal Reduced-Order Control | 44 |

| | | |
|-----|--|-----|
| 5. | Methods and Algorithms for Controller Design | 49 |
| 5.1 | Selection of Cost Functional Weighting Matrices and Covariance Matrices | 49 |
| 5.2 | Disturbance Cancellation and Inverse Optimal Control | 53 |
| 5.3 | Discrete-Time Controller Design..... | 58 |
| | 5.3A. Conversion of Cost and Covariance Matrices to Discrete-Time | 58 |
| | 5.3B. Computational Delay..... | 61 |
| 5.4 | Iterative Relaxation Algorithm for Solving Coupled Riccati/Lyapunov Equations | 62 |
| 5.5 | Homotopy Algorithm for Solving Optimal Projection Equations..... | 67 |
| 6. | Evaluation of Designs | 75 |
| 6.1 | Problem Description..... | 75 |
| 6.2 | Comparison of Tradeoff for Different Methods..... | 82 |
| | 6.2A. Uncertainty in Natural Frequency | 83 |
| | 6.2B. Uncertainty in Eigenvector | 86 |
| | 6.2C. Results for Different Factorizations..... | 88 |
| 6.3 | Itô vs. Stratonovich Noise..... | 93 |
| 6.4 | Reduced-Order Controller | 94 |
| 7. | Application to Simply Supported Plate | 101 |
| 7.1 | Stability Robustness Problem and Its Solution | 101 |
| 7.2 | Effectiveness of Reduced-Order Controllers | 104 |
| 8. | Conclusions | 111 |
| 8.1 | Robustness | 111 |
| 8.2 | Controller Order..... | 112 |
| 8.3 | Parameter Robust Reduced-Order Design..... | 113 |
| 8.4 | Directions for Further Study | 113 |
| | References | 114 |
| | Vita | 118 |

List of Figures

| | Page |
|--|---------|
| Figure 1.1: Nominal Performance for LQG vs. \mathcal{H}_∞ -Optimal Control | 7 |
| Figure 1.2: PRLQG Error Model | 10 |
| Figure 1.3: LQG/PRE Error Model | 10 |
| Figure 1.4: Time-Domain Auxiliary Input Models | 14 |
| Figure 1.5: Frequency-Domain Auxiliary Input Models..... | 15 |
| Figure 2.1: Model Used to Approximate Power Spectral Densities of Auxiliary Inputs | 20 |
| Figure 2.2: Plant Augmented with Noise Shaping Filters..... | 20 |
| Figure 6.1: Frequency Responses of Augmented Plant Components for 4-Mode Model..... | 81 |
| Figure 6.2: Steady State Performance and Stability Tradeoff for Uncertainty in Natural Frequency..... | 85 |
| Figure 6.3: Transient Performance and Stability Tradeoff for Uncertainty in Natural Frequency..... | 87 |
| Figure 6.4: Steady State Performance and Stability Tradeoff for Uncertainty in Eigenvector..... | 89 |
| Figure 6.5: Transient Performance and Stability Tradeoff for Uncertainty in Eigenvector..... | 90 |
| Figure 6.6: Comparative Performance for Several Different LQG/PRE Factorizations | 92 |
| Figure 6.7: Frequency Responses for Full- and Reduced-Order Compensators. | 96 |
| Figure 6.8: Disturbance Rejection for Full- and Reduced-Order Compensators | 97 |
| Figure 6.9: Frequency Response of Full- and Reduced-Order Compensators.... | 99 |
| Figure 6.10: Time Response for Full- and Reduced-Order Compensators | 100 |
| Figure 7.1: Performance/Stability Robustness Tradeoff for Uncertainty in ω_2 . | 103 |
| Figure 7.2: Robust Control — Mode 1 Response to 60 Hz Disturbance | 105 |
| Figure 7.3: Robust Control — Mode 2 Response to 60 Hz Disturbance | 106 |

| | |
|--|-----|
| Figure 7.4: Response of Full-Order Robust Controllers With and Without Time Delay | 108 |
| Figure 7.5: Robust Reduced-Order Control — Mode 1 Response to 60 Hz Disturbance | 109 |
| Figure 7.6: Robust Reduced-Order Control — Mode 2 Response to 60 Hz Disturbance | 110 |

List of Tables

| | Page |
|--|------|
| Table 6.1: Natural Frequencies and Damping Ratios for the First Nine Modes | 77 |
| Table 6.2: Controller Design Parameters Corresponding to Eight Different Factorizations | 91 |
| Table 6.3: LQ Cost and Disturbance Rejection vs. Controller Order | 95 |
| Table 7.1: Sample Rates for Full- and Reduced-Order Controllers | 107 |

0. Introduction

0.1 Purpose and Importance of This Work

The purpose of this research is to explore two new and related methods of designing compensators for the active control of uncertain systems. Both methods are designed to improve the robustness of a control system's stability and performance with respect to uncertainty in selected parameters of the state-space model of the plant. They accomplish this by modeling the effect of uncertainties as accurately as possible by fictitious noise sources while simultaneously reducing sensitivity by means of additional penalties in the cost functional. The first method constructs the noise model in the frequency domain, whereas the second formulates noise in the time domain. In addition, both incorporate optimal order reduction directly into the design. These methods are expected to have general application to plants with uncertain state-space parameters; however, they will be applied in this work to the problems involved in the suppression of disturbances in flexible structures, which suffer particularly from the difficulties of large plant uncertainty and large compensator order.

Modeling and identification of flexible structures, themselves, constitute a difficult problem and an active area of research (e.g., Balas and Doyle 1990). The prevalence of this type of research is an indicator that linear, finite-dimensional, time-invariant (LFDTI) models of flexible structures are examples of highly uncertain plants, and it points to the need for robust controllers for such systems.

Also, LFDTI models of flexible structures tend to be of very high order. Firstly, since flexible structures are infinite-dimensional, it is desirable to consider as many vibrational modes as possible in order to reduce the effect of unmodeled dynamics on the control system. Secondly, it is frequently necessary to augment the plant with additional dynamics in order to model disturbances more accurately or to meet more precise performance specifications. Disturbance modeling (e.g., Kwakernaak and Sivan 1972, sec. 1.11.4) is desirable when the disturbances are correlated and something is known about their frequency content. This allows the exogenous signals to be modeled by white noise processes (in the case of LQG-based designs), provided the disturbance dynamics are appended to the plant at the disturbance inputs. Likewise, frequency-shaped cost functionals (Gupta 1980) are used when the state and control weighting matrices are functions of frequency. This is accomplished in the LQG framework by

appending dynamics to the plant at the controlled-variable outputs. It has been shown (Sievers and von Flotow 1989) that these two extensions to LQG theory are duals and, in the single-input single-output (SISO) case, equivalents of one another. Also, the same two methods are used for analogous purposes in \mathcal{H}_∞ -based design (e.g., Doyle 1984).

In either case, an LQG or \mathcal{H}_∞ -optimal compensator, being of the same order as the augmented plant, must also be of high order. As a result, full-order controllers tend to be of very high order and therefore place a great computational burden on the real-time processor hardware.

0.2 Research Objectives

The proposed parameter robust reduced-order control methods are described in Chapter 1 and developed in Chapters 2 through 5. Then, the resulting designs are applied to a continuous-time FDLTI model of a simply supported rectangular plate (in Chapter 6), as well as to the actual hardware (in Chapter 7). The control system has one control input actuator, one disturbance input actuator, and twelve accelerometer sensors. The state-space model of the plate has been derived from modal frequency and modeshape data obtained by an identification procedure. This model is augmented by the dynamics associated with a control signal smoothing filter and a noise shaping filter designed to reflect the characteristics of a colored noise disturbance. Analysis has shown the stability of the closed-loop system to be most susceptible to errors in the natural frequencies and control input modeshape vector, and experimentation has confirmed this. Therefore, emphasis is placed on making the system less sensitive to errors in the parameters of the state-space model corresponding to these quantities.

There are three main objectives to this research. The first objective is to evaluate the efficiency of the performance/stability robustness tradeoff for the two proposed design methods. This is done by comparing compensators designed by the frequency-domain and time-domain noise modeling techniques (of Chapters 2 and 3, respectively) with those designed by two existing methods discussed in Chapter 1: LQG/PRE and multiplicative white noise modeling (without the auxiliary output modeling feature used in Chapter 3). First, a parameter range is specified for which the system must be stable. This parameter range is larger than the stability range obtained

by the standard LQG design. Second, full-order compensators are designed by all four techniques to just meet this stability robustness specification, using the same performance criterion and assumed exogenous noise covariances used by the LQG design. Finally, it is determined how much performance was sacrificed to attain this level of robustness by measuring the performance of the designs (i.e., the quadratic cost) over the stability range of interest. This method is applied to a simple 1-mode model of the plate in Chapter 6 to illustrate the design considerations involved. Uncertainties in both the control input modeshape vector (i.e., the controller actuator location) and the natural frequency are considered. In Chapter 7, a 4-mode model of the plate is analyzed to determine the cause of the poor stability properties of the actual hardware. Then the most suitable robust controller is selected and implemented on the plate.

The second objective is to investigate the extent to which the order of a standard LQG compensator can be reduced for FDLTI models of the rectangular plate. When one adds the states of a second-order smoothing filter and a noise shaping filter to the two states per vibrational mode of the plate, the high order of the plant becomes apparent, as does the need for a reduced-order compensator. The procedure is as follows. First, a full-order LQG compensator is designed to provide satisfactory nominal performance. Then reduced-order compensators are designed using the optimal projection equations, and it is determined how far the order can be reduced without significant degradation in performance. This is carried out in Chapter 6, and the results of implementation in hardware are described in Chapter 7.

The third and final objective is to combine the best modified LQG/PRE design method with the optimal projection equations to determine the overall merit of the new robust, minimal order design. This work is carried out in Chapter 7.

1. Motivation for This Study

The ideas developed in this work were motivated by the shortcomings of existing \mathcal{H}_2 - and \mathcal{H}_∞ -based techniques for rejecting disturbances in uncertain systems. In this chapter, standards are defined for the practical merit of a controller, and two approaches are proposed to better meet those standards than do existing methods. Then the specific contributions of this work are outlined.

1.1 Criteria for the Merit of Candidate Designs

The standards that will be used to judge and compare the effectiveness of various compensator designs are: 1) nominal disturbance rejection performance; 2) "efficiency" of tradeoff between performance and stability robustness; 3) compensator order; and 4) ability to reject both steady state and transient disturbances. What follows is a discussion of these criteria and the means by which they will be met.

A. Nominal Performance

The *performance* of a compensator will be measured by its ability to minimize the effect of unwanted exogenous signals on the prescribed controlled-variable vector, whose elements consist of weighted linear combinations of system states and controls. *Nominal performance* refers to the performance of a control system with all plant parameters equal to their assumed values. Two competing classes of performance optimality have been considered, LQG and \mathcal{H}_∞ . These two definitions of optimality differ in the way they model the exogenous signals and in the way they measure the magnitude of the resulting disturbance in the controlled variables.

In the traditional LQG formulation of the disturbance rejection problem, optimal performance is defined to be that which minimizes the cost functional

$$J = \lim_{T \rightarrow \infty} \frac{1}{2T} E \left\{ \int_{-T}^T [x^T(t)Qx(t) + u^T(t)Ru(t)]dt \right\} \quad (1.1)$$

in the presence of unit intensity, uncorrelated white process and sensor noise (v and n , respectively). For the purpose of easier comparison with \mathcal{H}_∞ -based methods, an equivalent formulation will be made in the frequency domain. Define the noise vector, w , and the controlled-variable vector, z , as follows:

$$w = \begin{bmatrix} v \\ n \end{bmatrix}, \quad z = \begin{bmatrix} Q^{1/2}x \\ R^{1/2}u \end{bmatrix} \quad (1.2)$$

and denote the closed-loop transfer function matrix from w to z by $H(s)$. Then the \mathcal{H}_2 -optimal compensator is that which minimizes the \mathcal{H}_2 norm of $H(s)$ (i.e., $\|H(s)\|_2$). The fact that the LQG and \mathcal{H}_2 definitions of optimal performance are equivalent are an immediate consequence of the following theorem:

Theorem. $J = \|H(s)\|_2^2$.

Proof. Using the subscript " T ", define time-truncated functions as follows:

$$\begin{aligned} x_T(t) &\triangleq x(t), & -T \leq t \leq T; \\ &\triangleq 0, & \text{otherwise} \end{aligned}$$

Then

$$\begin{aligned} J &\triangleq \lim_{T \rightarrow \infty} \frac{1}{2T} \mathbb{E} \left\{ \int_{-T}^T [x^T(t)Qx(t) + u^T(t)Ru(t)] dt \right\} \\ &= \lim_{T \rightarrow \infty} \frac{1}{2T} \mathbb{E} \left\{ \int_{-T}^T z^T(t)z(t) dt \right\} \\ &= \lim_{T \rightarrow \infty} \frac{1}{2T} \mathbb{E} \left\{ \int_{-\infty}^{\infty} z_T^T(t)z_T(t) dt \right\} \\ &= \lim_{T \rightarrow \infty} \frac{1}{2T} \mathbb{E} \left\{ \frac{1}{2\pi} \int_{-\infty}^{\infty} z_T^*(j\omega)z_T(j\omega) d\omega \right\} \\ &= \lim_{T \rightarrow \infty} \frac{1}{2T} \mathbb{E} \left\{ \frac{1}{2\pi} \int_{-\infty}^{\infty} \text{trace} [w_T^*(j\omega)H^*(j\omega)H(j\omega)w_T(j\omega)] d\omega \right\} \\ &= \frac{1}{2\pi} \int_{-\infty}^{\infty} \text{trace} \left[H(j\omega) \lim_{T \rightarrow \infty} \left\{ \frac{\mathbb{E} [w_T(j\omega)w_T^*(j\omega)]}{2T} \right\} H^*(j\omega) \right] d\omega \\ &= \frac{1}{2\pi} \int_{-\infty}^{\infty} \text{trace} [H^*(j\omega)H(j\omega)] d\omega \\ &= \|H(s)\|_2^2. \end{aligned}$$

In the proof of the theorem, use was made of the definition of the \mathcal{H}_2 operator norm (e.g., Dailey 1990), Parseval's Theorem, and the fact that the covariance of the

assumed disturbance vector, w , is the identity matrix.

In \mathcal{H}_∞ controller design, the same closed-loop transfer function is considered, and optimality of performance is achieved when its \mathcal{H}_∞ norm (i.e., $\|H(s)\|_\infty$) is minimized. However, the disturbances, w , and the controlled variables, z , are viewed differently. The \mathcal{H}_∞ operator norm is the induced L_2 norm, meaning

$$\|H(s)\|_\infty = \sup_{\|v(t)\|_2 = 1} \|z(t)\|_2 \quad (1.3)$$

Therefore, w and z are considered to be L_2 (i.e., square integrable) signals. This expression for the \mathcal{H}_∞ norm reveals a possible drawback to \mathcal{H}_∞ -based control methods for disturbance rejection. Since the \mathcal{H}_∞ controller is designed for the worst case unity-norm L_2 disturbance, no consideration is given to the relative intensities of multiple noise sources. The worst case disturbance may be one whose entire power is delivered to a single process noise or sensor noise port. LQG-based methods, on the other hand, model the noise intensity at each port separately.

Another expression for the \mathcal{H}_∞ norm is given by its definition,

$$\|H(s)\|_\infty \triangleq \sup_{\omega} \bar{\sigma}[H(j\omega)] \quad (1.4)$$

where $\bar{\sigma}$ denotes the maximum singular value. This expression shows that the \mathcal{H}_∞ -optimal compensator is the one which minimizes the *worst case* disturbance rejection over all frequencies. In contrast, the LQG compensator provides better disturbance rejection *averaged* over all frequencies. This principle is illustrated in the singular value plot of a closed-loop vibrational system in Figure 1.1. Using the same state and control weightings, Q and R , an LQG and an \mathcal{H}_∞ compensator were designed to reject disturbances in an eight-mode model of a flexible structure. The two curves indicate the maximum singular value of the transfer function matrix, $H(s)$, for the two different compensator designs. The \mathcal{H}_∞ compensator provided slightly better worst case performance, but at the cost of significantly worse performance averaged over the entire frequency range.

Because of these two potential problems associated with disturbance rejection in \mathcal{H}_∞ -based designs (i.e., less precise noise modeling and less practical definition of

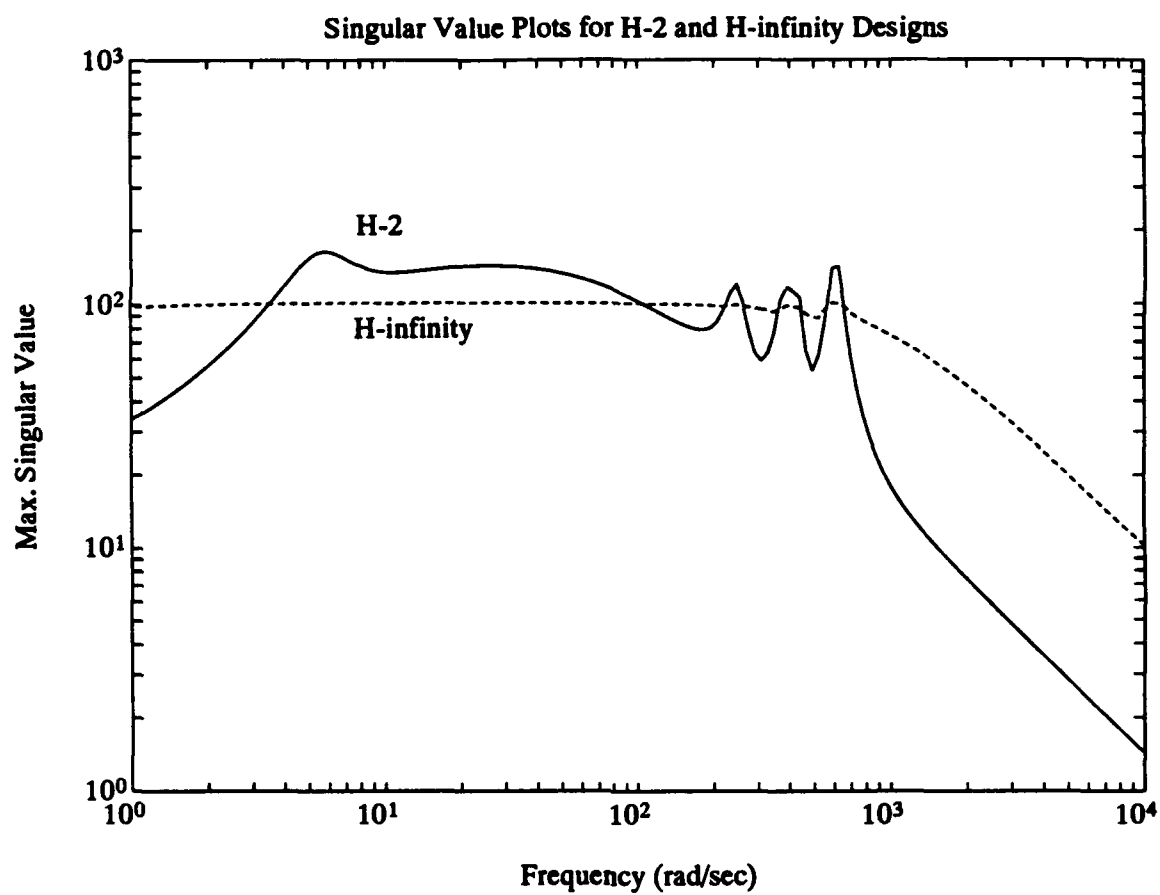


Figure 1.1: Nominal Performance for LQG vs. \mathcal{H}_∞ -Optimal Control

optimality) preferential consideration was given to improving LQG-based design methods in the expectation that they would, by conventional measures, provide significantly better nominal performance.

B. Performance/Stability Robustness Tradeoff

For uncertain systems, \mathcal{H}_∞ -based methods, such as μ -synthesis (Doyle 1985), have the advantage that plant uncertainties with singular value bounds can be incorporated directly into the design, without the need for fictitious noise sources to model the uncertainties. However, these methods also have serious limitations, as illustrated in their recent application to a flight control problem (Doyle, Lenz, and Packard 1987). Firstly, \mathcal{H}_∞ controllers do not provide optimal performance and stability robustness simultaneously. For example, an \mathcal{H}_∞ controller designed for performance may yield very poor stability margins. Secondly, the μ -synthesis technique corrects this problem by attempting an optimal tradeoff between performance and stability robustness, but the resulting controllers tend to be of very high order. Therefore, ad hoc controller and/or model reduction schemes are required to reduce the controller order to a practical size.

An LQG-based method can avoid these problems, provided that its noise models adequately represent the uncertainties without greatly increasing the order of the compensator. For this reason, the main thrust of this work is to improve on an existing LQG-based robust control method by more accurately modeling the plant uncertainties by means of more suitable fictitious noise sources. The purpose of doing so is to give the resulting compensators the same degree of stability robustness with a smaller sacrifice in performance. The degree to which this occurs will be referred to as the "efficiency" of the performance/stability robustness tradeoff.

Tahk and Speyer (1987) introduced a modified LQG synthesis procedure, later called parameter robust LQG (PRLQG), in which perturbations in the elements of the A -matrix are modeled by a fictitious internal feedback loop. Given a nominal system matrix, A , and the perturbed matrix, \hat{A} , define the perturbation matrix by $\Delta A = \hat{A} - A$. Then factor it as follows: $\Delta A = MLN$, where M and N are of full rank (to minimize their dimensions) and L is without loss of generality a diagonal matrix, whose diagonal elements reflect the (unknown) magnitude of the perturbations. An accurate model of the perturbed system results after the addition of this internal

feedback loop, as shown in Figure 1.2. Now, given the nominal system,

$$\begin{aligned}\dot{x} &= Ax + Bu + G_1 w \\ y &= Cx + G_2 w\end{aligned}\tag{1.5}$$

the perturbed system is described by

$$\begin{aligned}\dot{x} &= Ax + Bu + G_1 w + Mw_a \\ y &= Cx + G_2 w \\ z_a &= Nx \\ w_a &= Lz_a\end{aligned}\tag{1.6}$$

where w_a and z_a are auxiliary input and output variables, respectively. Intuitively, one may suspect that the robustness of the system to the prescribed parameter variations would be enhanced if w_a were modeled as white noise, thereby adding μMM^T (for some scalar μ) to the process noise covariance. Actually, Tahk and Speyer showed that under certain minimum phase and similarity conditions, the robustness of an LQ regulator is recovered asymptotically as $\mu \rightarrow \infty$. Note that this robustness recovery technique is identical to LQG/LTR for uncertainties at the input (Doyle and Stein 1981), provided $M = B$. However, the PRLQG method has the advantage that M contains partial information as to the structure of the perturbation, ΔA . In fact, Tahk and Speyer showed that LQG/LTR fails to asymptotically desensitize the estimator (when PRLQG succeeds) for certain structures of ΔA , since B does not necessarily pass a crucial similarity condition with respect to the matrix ΔA , namely that $\text{span}\{M\} \subset \text{span}\{B\}$ for some MLN -factorization of ΔA . Vibration control examples have demonstrated that in such cases LQG/LTR may provide relatively poor robustness to parameter uncertainties (Tahk and Speyer 1989). PRLQG also includes the dual of the above procedure — analogous to LQG/LTR for uncertainties at the output. In the dual procedure, z_a is treated as an auxiliary controlled variable, and the term $\rho N^T N$ (for some scalar ρ) is added to the state weighting matrix in the regulator design. This procedure also makes sense intuitively, as it seems reasonable to penalize the auxiliary output variable in the cost functional in order to minimize the magnitude of the auxiliary inputs that result. Combining the two procedures asymptotically yields “absolute robustness” (i.e., stability robustness to parameter uncertainties of arbitrary

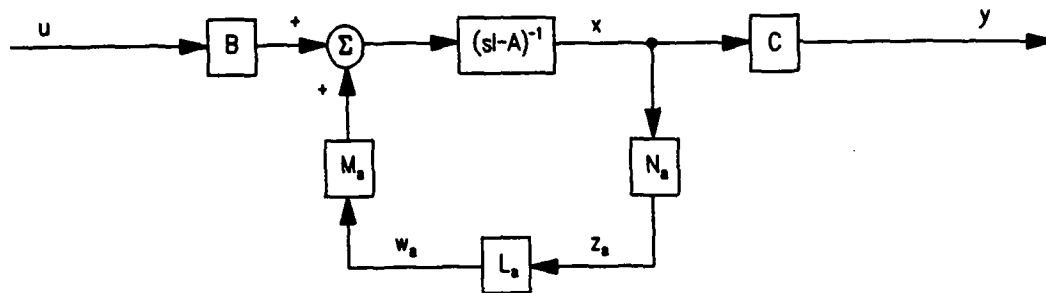


Figure 1.2: PRLQG Error Model

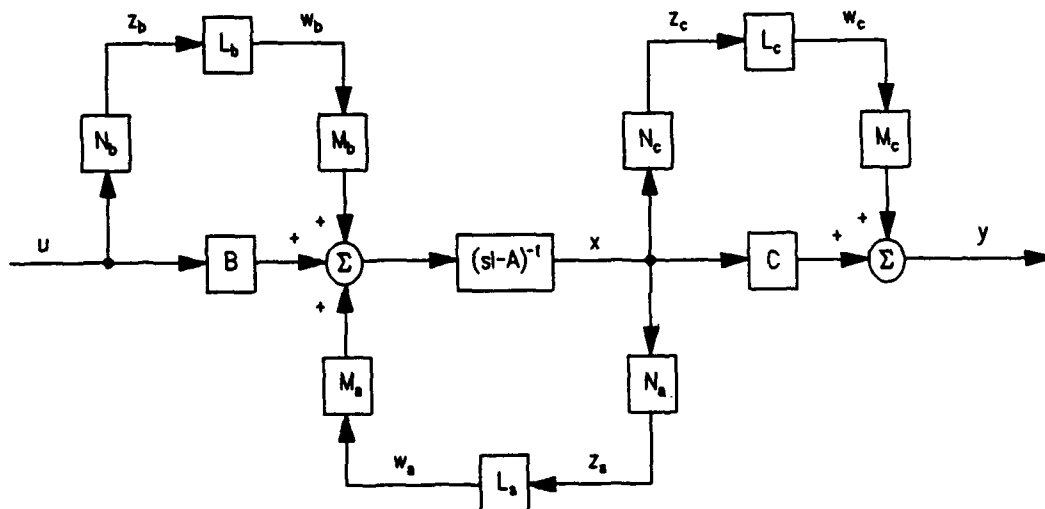


Figure 1.3: LQG/PRE Error Model

magnitude) as $\mu, \rho \rightarrow \infty$.

Lin (1989) studied the combined effect of both procedures in more detail and also extended the method to include parameter variations in the B - and C -matrices in a way that is less cumbersome than that suggested by Tahk and Speyer. This extended method was termed LQG/PRE (standing for Linear Quadratic Gaussian design with Parameter Robustness Enhancement). The resulting model of the plant, with perturbations in A , B , and C is shown in Figure 1.3. The system equations are:

$$\begin{aligned} \dot{x} &= Ax + Bu + G_1 w + M_a w_a + M_b w_b \\ y &= Cx + G_2 w + M_c w_c \\ z_a &= N_a x, \quad z_b = N_b u, \quad z_c = N_c x \\ w_a &= L_a z_a, \quad w_b = L_b z_b, \quad w_c = L_c z_c \end{aligned} \quad (1.7)$$

where

$$\Delta A \triangleq M_a L_a N_a, \quad \Delta B \triangleq M_b L_b N_b, \quad \Delta C \triangleq M_c L_c N_c \quad (1.8)$$

Given the nominal regulator weighting matrices, Q and R , and the nominal Kalman filter noise covariances, Q_f and R_f , this leads to the following modified LQG design matrices (denoted with carats):

$$\begin{aligned} \hat{Q} &= Q + \rho_a N_a^T N_a + \rho_c N_c^T N_c \\ \hat{R} &= R + \rho_b N_b^T N_b \\ \hat{Q}_f &= Q_f + \mu_a M_a M_a^T + \mu_b M_b M_b^T \\ \hat{R}_f &= R_f + \mu_c M_c M_c^T \end{aligned} \quad (1.9)$$

The main weakness of the LQG/PRE scheme seems to be the fact that it models the auxiliary input variables as white noise. In order to reflect reality, these fictitious input signals should generally be modeled quite differently. For example, suppose there is a single uncertain, but constant, parameter in A (i.e., w_a is scalar, and ΔA is constant). Then a true model of the perturbed system would require w_a to be proportional to one of the states, say x_i (through $w_a = L_a N_a x$). Since the sign and magnitude of the perturbation is unknown (i.e., L_a unknown), w_a cannot be modeled faithfully; however, intuitively, it seems its dynamics should more closely resemble the dynamic behavior of x_i rather than that of a constant-intensity white noise variable.

Considerable research has already been conducted that will be applied to constructing a more meaningful model of the auxiliary inputs. This research was done

outside the LQG/PRE-framework, and did not include the related problem of penalizing auxiliary outputs. Using stochastic differential equation theory, Wonham (1967, 1968) developed modified Riccati equations for both LQ regulator and Kalman filter design for systems with "state-dependent noise". State-dependent noise arises in systems with state equations of the form

$$\dot{x} = Ax + Bu + G(x)w \quad (1.10)$$

where $G(x)$ is some function of the state, and w is a (vector) white noise process. In the above example, we would model w_a as state-dependent noise, which would make it white noise with time-varying intensity proportional to x_i . Equivalently, one could conceptualize ΔA as a matrix whose elements are white noise processes.

Hyland (Hyland and Madiwale 1981, Hyland 1982), motivated by the maximum entropy principle and the concept of ΔA as a matrix of white noise processes, modeled uncertainties in the A -matrix by adding state-dependent noise to the plant model. In doing so, he developed the equations for the full LQG compensator for systems with state-dependent noise. For such systems, the separation principle no longer holds. The regulator and filter design require the solution not of two uncoupled Riccati equations, but of four coupled equations — two modified Riccati equations and two modified Lyapunov equations. Later, Bernstein and Hyland (1988a) extended this result to apply to systems with state-, control-, and measurement-dependent noise, effectively allowing parameter perturbations in all three system matrices (ΔA , ΔB , and ΔC , respectively) to be modeled as white noise processes. The design process is computationally more difficult than that of simple LQG in that it requires the iterative solution of four coupled matrix equations, but notably the optimal compensator is of the same order as the plant. That is, adding state-, control-, and measurement-dependent noise (henceforth referred to collectively as *multiplicative white noise*) to a system involves no increase in compensator order.

As stated at the beginning of this section, this research involves modeling fictitious noise sources more accurately in an LQG-type problem. Specifically, the LQG/PRE approach will be adopted, with modifications that are intended to improve the efficiency of the performance/stability robustness tradeoff by remodeling the auxiliary input signals such that they more closely reflect reality. Two different fictitious noise models will be investigated, one motivated by consideration of the time-

domain knowledge of the auxiliary input signals, and the other by frequency-domain knowledge.

The time-domain method consists of using multiplicative white noise in place of simple white noise to model the auxiliary input signals. Recalling the simple example mentioned above, the motivation for this approach is illustrated in Figure 1.4. These plots were derived from a two-mode simulation of the rectangular plate apparatus described in section 1.3. A true constant parameter uncertainty in the A -matrix would be compensated for optimally if w_a were modeled perfectly (i.e., $w_a = kx_i$, where the constant k is determined by the sign and magnitude of the parameter variation), as shown in Figure 1.4a. However, since k is unknown, this model is impossible to implement. The LQG/PRE method uses constant-intensity white noise to model w_a (say, $w_a = w$), as in Figure 1.4b. But the multiplicative white noise approach will use state-dependent noise to take advantage of the fluctuations in x_i (i.e., $w_a = x_i w$), as shown in Figure 1.4c. Presumably, the extent to which this noise model improves the LQG/PRE design will depend on the degree to which the amplitude of x_i varies over time. In particular, the multiplicative noise model should provide a greater improvement for transient noise rejection than for steady-state (i.e., constant-intensity) noise rejection. This modification of LQG/PRE requires solution of the coupled Riccati and Lyapunov equations, but does not result in an increase in compensator order.

The frequency-domain method involves an attempt to match the true auxiliary inputs in the frequency domain. Time-domain matching suffers from the fact that the sign of the parameter perturbations is unknown. As a result, a very broad band signal is used in an attempt to cover both possibilities at every time instant. However, our knowledge of the *frequency* content of the auxiliary inputs is limited only by our knowledge of the disturbance's frequency content and of the closed-loop system dynamics. So, given an assumed disturbance model, the true auxiliary inputs can be modeled fairly accurately in the frequency domain by closing the loop with an LQG compensator which gives the desired nominal performance and measuring the frequency response of w_a , w_b , and w_c (see Figure 1.3). Therefore, the frequency-domain behavior of the auxiliary inputs will be approximated by passing fictitious white noise signals through suitable frequency-shaping filters. The effect is demonstrated in the frequency spectrum plots of Figure 1.5, generated from the same data used in Figure 1.4. The frequency-shaped noise model, produced by means of a second-order filter, provides a much better frequency-domain replica of the true signal ($w_a = kx_i$) than does a white

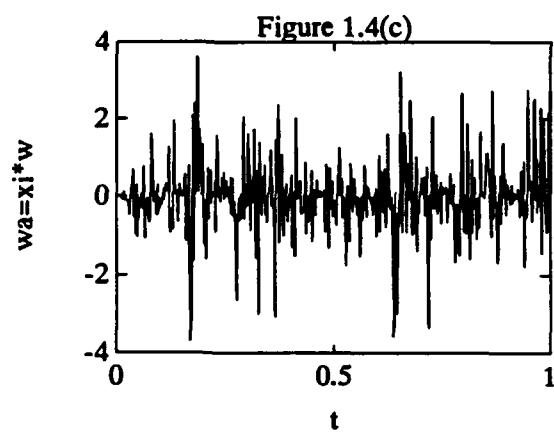
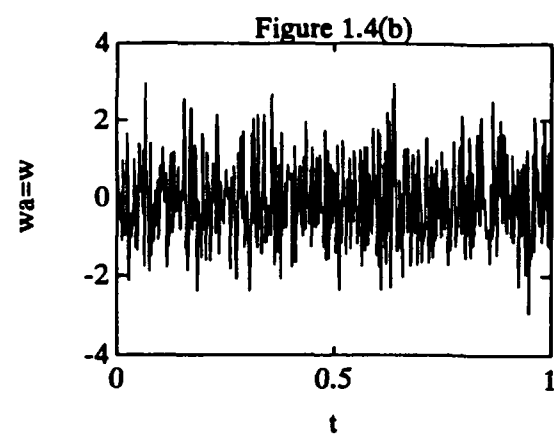
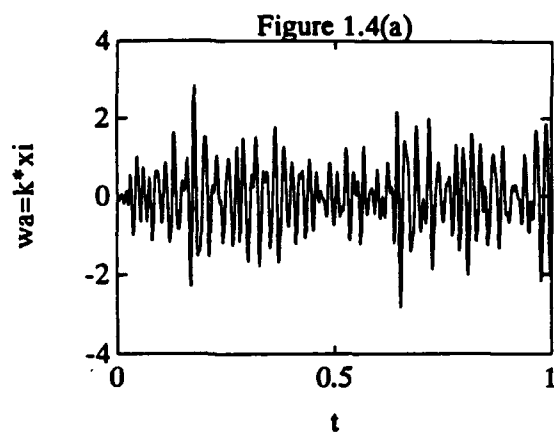


Figure 1.4: Time-Domain Auxiliary Input Models

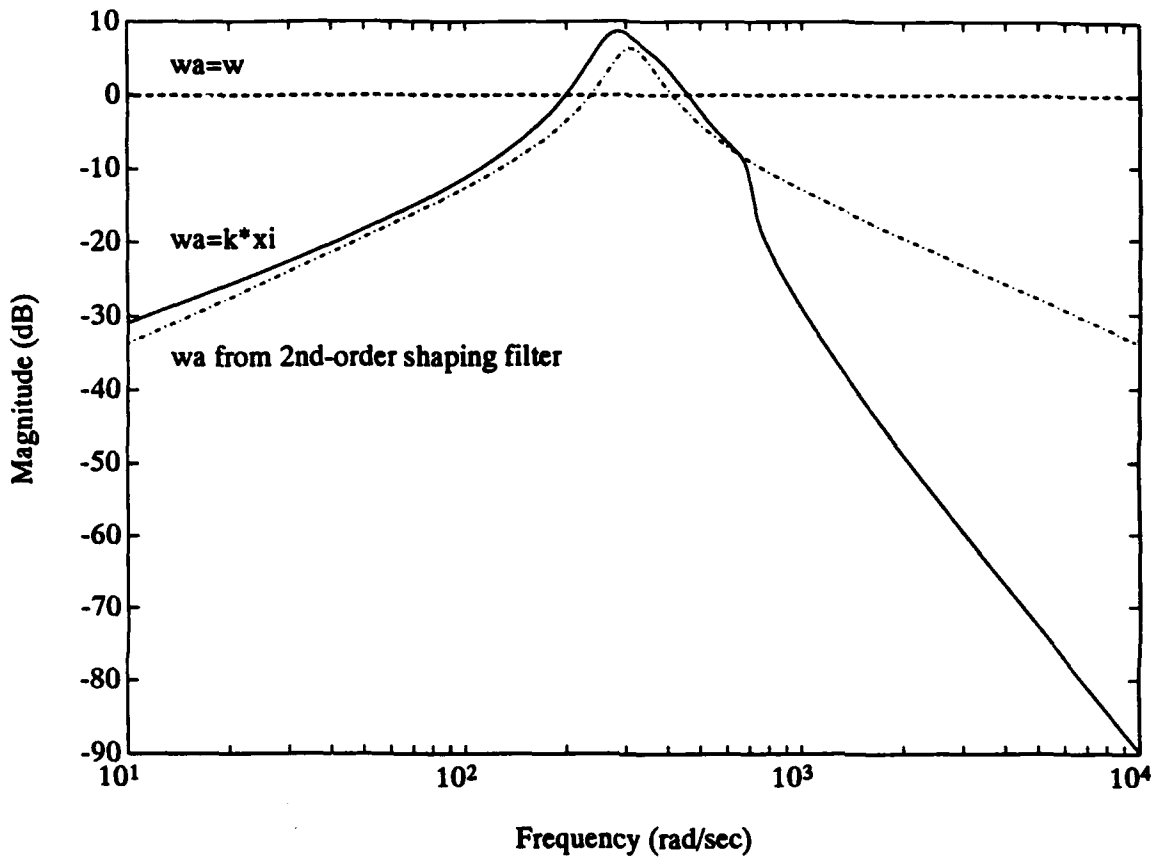


Figure 1.5: Frequency-Domain Auxiliary Input Models

noise model ($w_a = w$). As with time-domain matching, the correct magnitude of the auxiliary inputs cannot be duplicated, but the intensity of the noise signals used to generate the noise models may be adjusted as design parameters. This modification of LQG/PRE does not require solution of the coupled Riccati and Lyapunov equations; however, it effectively increases the size of a full-order compensator, since it involves augmentation of the plant model with additional dynamics.

C. Compensator Order

Many methods are available to reduce compensator order, but they fall into three basic categories — *model reduction*, *controller reduction*, and *direct design* (Anderson and Liu 1989). Model and controller reduction may be accomplished by such methods as balanced truncation (Moore 1981) or Hankel norm approximation (Glover 1984), but the optimality of the compensator (for a given controller order) — and sometimes even stability — is preserved only by designing a reduced-order controller *directly* from a full-order plant model. The equations for the direct design of LQG reduced-order controllers were developed by Kwakernaak and Sivan (1972, sec. 5.7), among others. However, the solution of these equations requires a gradient search on the many free parameters of the controller state-space model. A far more practical method was developed by Hyland and Bernstein (1984), involving solution of the *optimal projection equations* — a coupled system of two modified Riccati and two modified Lyapunov equations, similar to those mentioned above in the discussion on stochastic control with multiplicative white noise. The Lyapunov-type equations are analogous to the Lyapunov equations required in solving the balanced truncation of the plant or controller, but they are coupled with the modified Riccati equations by means of an optimal projection matrix when the controller is less than full-order. This coupling demonstrates the fact that balancing and controller design cannot be performed separately to obtain an optimal reduced-order controller.

The suggested modifications to LQG/PRE easily fit into the framework of optimal reduced-order compensators, with minor modifications to the optimal projection equations. This is an important feature, since the degree to which controller reduction is possible may very well determine whether modern control applied to flexible structures can be implemented.

D. Steady State, Transient Disturbance Rejection

When sensors are available to measure incoming disturbances, feedforward control becomes an alternative to feedback control. The main advantage of feedforward control is that the compensator does not need to, in effect, model the plant dynamics. Therefore, feedforward control appears to be an attractive alternative when small compensator order is important. However, experience has shown that the advantage in compensator order can be significantly reduced if the feedback controller is designed by optimal projection. Also, feedback controllers can provide added damping to the plant and therefore provide the ability to reject transient disturbances. For these reasons, only feedback control techniques are studied here.

1.2 Contributions

This research has a number of contributions. Firstly, the internal feedback loop (IFL) modeling principle of Tahk and Speyer is generalized and fully exploited to maximize its potential. Secondly, the concept of multiplicative white noise is given broader application by means of a new interpretation, and a complete derivation is given of the controller design equations from first principles. Thirdly, implementation concerns are thoroughly discussed, and explicit algorithms are developed that have application beyond this study. Lastly, the validity of the controller design techniques is demonstrated by their implementation on flexible structure hardware.

The original IFL modeling technique, used by Tahk and Speyer for PRLQG and by Lin for LQG/PRE, was restricted to a white noise model for the auxiliary inputs and resulted only in full-order controllers. In this research, it is shown that a reliance on white noise models may result in relatively poor performance. Frequency-shaped noise and multiplicative white noise models are developed to improve the flexibility of this LQG-based design method. Also, optimal order reduction is incorporated into this framework for the first time. The tremendous savings in controller duty cycle that result are demonstrated, suggesting that the addition of this feature may be critical to the ability of these parameter-robust controllers to be implemented.

The controller design equations for plant models with multiplicative white noise have already been derived for application to a less general and differently motivated robustness problem. However, derivations published up to now have been incomplete,

frequently presuming knowledge of stochastic differential equations and omitting nontrivial steps and clarifications which fully explain the applicability of the design equations. This work attempts to fill that void by supplying an entire and unbroken derivation, complete with surrounding discussion and explicit references to readily available sources.

In the implementation phase of the design, a number of decisions must be made which affect the performance of the control system. The procedures that were used to resolve these problems are discussed in detail, and explicit algorithms are developed for the solution of the coupled Riccati and Lyapunov equations of Chapters 3 and 4.

The application of the parameter-robust and reduced-order controller designs to the control of an actual flexible structure demonstrate their ability to solve real problems and allow us to quantify the performance of these designs. Sacrifices in performance due to robustness enhancement and controller reduction are measured and limitations are discovered as to the maximum amount of controller reduction possible.

2. Frequency-Shaped Noise

2.1 Problem Statement

The frequency-domain method of auxiliary input modeling attempts to replicate the frequency-domain behavior of the (unknown) auxiliary input signals by means of a frequency-shaped noise model. Rather than using white noise (as does LQG/PRE) to model the auxiliary input signals w_a , w_b , and w_c (see Figure 1.3), a much more precise determination is made of the actual power spectral density of these signals. The resulting auxiliary noise model is then combined with a cost functional penalty on the auxiliary output signals — z_a , z_b , and z_c — to form a new method of parameter-robust controller design. The noise modeling phase of this design method consists of 1) finding an approximation for the power spectra of the auxiliary input signals, 2) designing a noise shaping filter to warp the power spectrum of white noise into this shape, and 3) appending the shaping filter dynamics to the plant.

Referring to the LQG/PRE model of the plant in equations (1.7) and Figure 1.3, we see that the actual power spectra of the auxiliary inputs cannot be found, because these signals are a function of the unknown diagonal matrices L_a , L_b , and L_c , and of the compensator matrix triple, $\{A_c, B_c, C_c\}$, which is yet to be designed. However, a very good approximation to the power spectra may still be found. For relatively small perturbations in the A -, B -, and C -matrices of the plant, compared to the nominal values, the feedback and feedforward loops containing the auxiliary signals have little effect on the frequency content of the states and controls, upon which the auxiliary inputs depend. Therefore, we may safely break these loops at the auxiliary inputs (see Figure 2.1) for the purpose of approximating the power spectra. Now, the unknown diagonal L -matrices only determine the sign and magnitude of the auxiliary inputs, not their frequency content, and the elements of the vectors w_a , w_b , and w_c are proportional to those of z_a , z_b , and z_c , respectively. Therefore, the task of finding the power spectral densities of the auxiliary input signals is equivalent to finding the power spectra of the auxiliary outputs, modulo some proportionality constants. The values of those constants are not important, because the relative noise intensities of the auxiliary inputs must later be adjusted in order to provide the desired amount parameter robustness. The problem of not knowing the compensator matrices a priori can be treated by designing a standard LQG compensator based on the nominal parameter values of the plant and using it to calculate power spectra. Experience has shown that the LQG

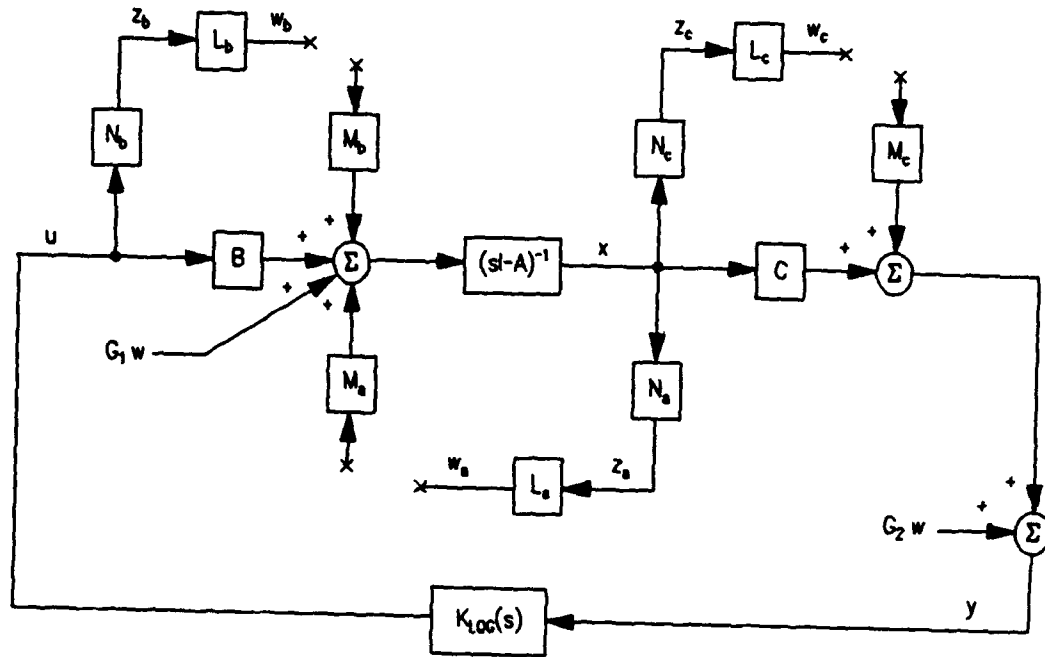


Figure 2.1: Model Used to Approximate Power Spectral Densities of Auxiliary Inputs

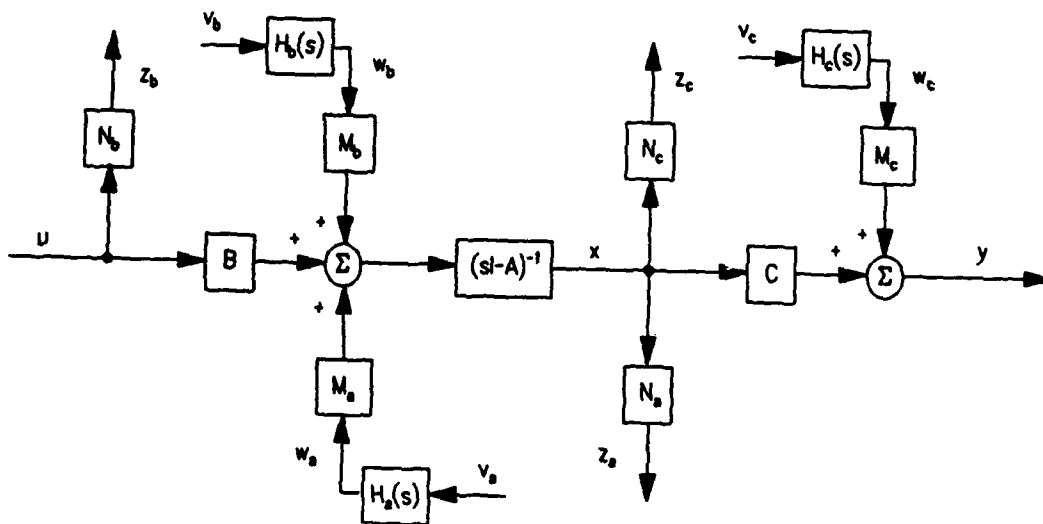


Figure 2.2: Plant Augmented with Noise Shaping Filters

compensator can provide an excellent substitute for this purpose. If greater accuracy were required, one could use an iterative process of alternately designing a compensator and approximating power spectra until a satisfactory noise model were found.

2.2 Noise Shaping Filter Design

Figure 2.1 illustrates the resulting model used to obtain approximations of the power spectral densities of the auxiliary input signals. The corresponding state-space model of the n -th order plant, n -th order compensator, and auxiliary inputs (assuming the L -matrices are all identity matrices) is given by

$$\dot{x}(t) = Ax(t) + Bu(t) + G_1w(t) \quad (2.1)$$

$$y(t) = Cx(t) + G_2w(t) \quad (2.2)$$

$$x_c(t) = A_c x_c(t) + B_c y(t) \quad (2.3)$$

$$u(t) = C_c x_c(t) \quad (2.4)$$

$$w_a(t) = N_a x(t), \quad w_b(t) = N_b u(t), \quad w_c(t) = N_c x(t) \quad (2.5)$$

It is assumed without loss of generality that the exogenous noise signal, $w(t)$, is a vector of mutually independent, unit intensity white noise processes. Included in $w(t)$ are both the process noise and sensor noise. The closed-loop system with input w and outputs w_a , w_b , and w_c is described by

$$\dot{\tilde{x}}(t) = \tilde{A}\tilde{x}(t) + \tilde{G}w(t) \quad (2.6)$$

$$w_a(t) = [N_a \quad 0] \tilde{x}(t) \quad (2.7)$$

$$w_b(t) = [0 \quad N_b C_c] \tilde{x}(t) \quad (2.8)$$

$$w_c(t) = [N_c \quad 0] \tilde{x}(t) \quad (2.9)$$

where

$$\tilde{x}(t) \triangleq \begin{bmatrix} x(t) \\ x_c(t) \end{bmatrix}, \quad \tilde{A} \triangleq \begin{bmatrix} A & BC_c \\ B_c C & A_c \end{bmatrix}, \quad \tilde{G} \triangleq \begin{bmatrix} G_1 \\ B_c G_2 \end{bmatrix} \quad (2.10)$$

From this closed-loop model and the assumptions on w , the power spectral densities of the elements of w_a , w_b , and w_c may be derived one at a time. Let us say, arbitrarily, that we are interested in w_a and that it is a scalar. The following procedure is easily repeated for all other existing auxiliary inputs. Then we have the frequency-domain description of w_a ,

$$w_a = [N_a \ 0](sI - \tilde{A})^{-1}\tilde{G}w \quad (2.11)$$

$$\triangleq H(s)w$$

In general, then, $H(s)$ is a $2n$ -th order, multi-input, single-output transfer function matrix. If the plant is of sufficiently low order, we could stop here and use $H(s)$ as the noise shaping filter to be appended to the plant. Normally, however, this would not be practical, so a lower-order filter will be found, which is single-input, single-output. One might suspect that a low-order balanced truncation approximation to $H(s)$ would provide a good substitute, but examination of their comparative frequency responses for particular examples has shown that there is a much more effective method of designing an accurate noise shaping filter.

First we find an expression for the power spectral density, $S_{w_a w_a}(\omega)$, of w_a . For a vector exogenous noise process w , the definition of $H(s)$ in (2.11) leads to (Maciejowski 1989, p. 98):

$$\begin{aligned} S_{w_a w_a}(\omega) &= H(\omega)S_{ww}(\omega)H^*(\omega) \\ &= H(\omega)H^*(\omega) \end{aligned} \quad (2.13)$$

where $H^*(\omega)$ is the complex conjugate transpose of the transfer function matrix from w to w_a . A suitable shaping filter designed to yield this power spectrum may be found by deriving its magnitude frequency response from the information in (2.13) and curve fitting a transfer function to give that response. The desired frequency response of our shaping filter transfer function, $H_s(\omega)$, is

$$|H_s(\omega)| = \sqrt{H(\omega)H^*(\omega)} \quad (2.14)$$

In order to hold down the controller order, the magnitude response in (2.14) should be fit using a shaping filter of as low order as possible.

2.3 Controller Design Summary

Once the shaping filters to produce auxiliary inputs w_a , w_b , and w_c have been found — say, $H_a(\omega)$, $H_b(\omega)$, and $H_c(\omega)$, respectively — they may be appended to the plant, as shown in Figure 2.2. Define the state-space models of the shaping filters by the following identities,

$$H_a(\omega) = C_{wa}(j\omega I - A_{wa})^{-1}B_{wa} + D_{wa} \quad (2.15)$$

$$H_b(\omega) = C_{wb}(j\omega I - A_{wb})^{-1}B_{wb} + D_{wb}$$

$$H_c(\omega) = C_{wc}(j\omega I - A_{wc})^{-1}B_{wc} + D_{wc}$$

Then the state-space model of the plant, (2.1) and (2.2), is augmented so that we have,

$$\begin{aligned} \frac{d}{dt} \begin{bmatrix} x \\ x_a \\ x_b \\ x_c \end{bmatrix} &= \begin{bmatrix} A & M_a C_{wa} & M_b C_{wb} & 0 \\ 0 & A_{wa} & 0 & 0 \\ 0 & 0 & A_{wb} & 0 \\ 0 & 0 & 0 & A_{wc} \end{bmatrix} \begin{bmatrix} x \\ x_a \\ x_b \\ x_c \end{bmatrix} + \begin{bmatrix} B \\ 0 \\ 0 \\ 0 \end{bmatrix} u + \begin{bmatrix} G_1 & M_a D_{wa} & M_b D_{wb} & 0 \\ 0 & B_{wa} & 0 & 0 \\ 0 & 0 & B_{wb} & 0 \\ 0 & 0 & 0 & B_{wc} \end{bmatrix} \begin{bmatrix} w \\ v_a \\ v_b \\ v_c \end{bmatrix} \\ y &= \begin{bmatrix} C & 0 & 0 & M_c C_{wc} \end{bmatrix} \begin{bmatrix} x \\ x_a \\ x_b \\ x_c \end{bmatrix} + Du + \begin{bmatrix} G_2 & 0 & 0 & M_c D_{wc} \end{bmatrix} \begin{bmatrix} w \\ v_a \\ v_b \\ v_c \end{bmatrix} \end{aligned} \quad (2.16)$$

The supplementary independent white noise variables v_a , v_b , and v_c are treated as exogenous noise sources, and there are as many of these additional noise sources in the augmented model as there are independent parameter uncertainties in the plant.

After augmenting the plant with the noise shaping filter dynamics as above, the rest of the frequency-domain method is the same as LQG/PRE. The intensities of the auxiliary inputs and the weights on the auxiliary outputs are adjusted to provide the desired amount of parameter robustness. Then a standard LQG compensator is designed based on the augmented model with modified noise intensities and cost functional.

3. Multiplicative White Noise

3.1 Problem Statement

The time-domain method of auxiliary input modeling uses artificial noise of time-varying intensity to account for the time-varying amplitude of the actual auxiliary input signals. Since the amplitude of the true signals is proportional to a known linear combination of the states or controls, a multiplicative white noise model provides the desired proportionate noise intensity as a function of time. As in Chapter 2, the auxiliary input model is supplemented with a cost functional penalty on the auxiliary outputs. The time-domain method without the added flexibility of auxiliary output penalties is equivalent to the compensator design method of Bernstein and Hyland (e.g., 1988a) for multiplicative white noise for uncertain systems, although the motivation here is different. Since the optimal linear quadratic compensator for the resulting plant model cannot be found by standard LQG techniques, a more lengthy development is required here than was necessary for the frequency-domain method of Chapter 2.

Modeling of the auxiliary input signals by multiplicative white noise requires modification of the state and/or output equations of the plant. For the general case of uncertainties in the A -, B -, and C -matrices, we have the n -th order system,

$$dx_t = Ax_t dt + \sum_{i=1}^p \gamma_i A_i x_t dv_{it} + Bu_t dt + \sum_{i=1}^p \gamma_i B_i u_t dv_{it} + G_1 d\beta_t \quad (3.1)$$

$$dy_t = Cx_t dt + \sum_{i=1}^p \gamma_i C_i x_t dv_{it} + G_2 d\beta_t \quad (3.2)$$

These equations are written in differential form, because the development of the LQ optimal compensator for systems with multiplicative white noise requires the use of a more rigorous form of stochastic differential equation theory. The theory makes subtle distinctions in the interpretation of noise processes, which will be reviewed briefly in the next section. Such precise definitions are inconsistent with the use of the concept of Gaussian white noise, which, strictly speaking, is not well defined in the more rigorous framework. Therefore, all of the noise variables in this chapter will be written as differentials of *Wiener processes*, also known as *Brownian motion processes* and loosely referred to as the "integral of Gaussian white noise". Note also that variables which are a function of time are identified as such by a " t " in the subscript. This

compact notation is used throughout the chapter in order to prevent the equations from becoming too cumbersome.

The model of equations (3.1) and (3.2) represents a system with p independent auxiliary input signals, arising from p independent parameter uncertainties. The v_{it} ($i = 1, 2, \dots, p$) are scalar independent Wiener processes, with $E\{dv_{it}^2\} = \gamma_i dt$, and β_t is a vector Wiener process with $E\{d\beta_t d\beta_t^T\} = V dt$. As an illustrative example, consider the following 1-mode damped oscillator with acceleration output and two independent auxiliary input signals.

$$dx_t = \begin{bmatrix} 0 & 1 \\ -\omega^2 & -2\zeta\omega \end{bmatrix} x_t dt + \begin{bmatrix} 0 & 0 \\ 0 & \delta_1 \end{bmatrix} x_t dv_{1t} + \begin{bmatrix} 0 \\ b \end{bmatrix} u_t + \begin{bmatrix} 0 \\ \delta_2 \end{bmatrix} u_t dv_{2t} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} d\beta_t \quad (3.3)$$

$$dy_t = \begin{bmatrix} -\omega^2 & -2\zeta\omega \end{bmatrix} x_t dt + \begin{bmatrix} 0 & \delta_1 \end{bmatrix} x_t dv_{1t} + \begin{bmatrix} 0 & 1 \end{bmatrix} d\beta_t \quad (3.4)$$

An uncertainty in the damping parameter, $-2\zeta\omega$, must occur in both the A - and C -matrices. Since these parameter uncertainties are necessarily the same, they comprise a single independent uncertainty and utilize the same noise process, dv_{1t} . The resulting auxiliary inputs consist of (1) *state-dependent noise* (noise with intensity proportional to the states and entering at the process noise port) and (2) *measurement-dependent noise* (noise with intensity proportional to the states and entering at the sensor noise port). An uncertainty in the B -matrix will generally be independent of the damping parameter uncertainty and is provided for by an independent noise process, v_{2t} , yielding an auxiliary input which is *control-dependent noise* (noise with intensity proportional to the input and entering at the process noise port). The positive scalar constants δ_1 and δ_2 are design parameters which effectively adjust the intensities of the auxiliary input noise signals according to the magnitude of the uncertainties present.

Given the n_c -th order state-space model of the compensator,

$$dx_{ct} = A_c x_{ct} dt + B_c dy_t \quad (3.5)$$

$$u_t = C_c x_{ct} \quad (3.6)$$

where $n_c = n$ for a full-order controller, we can represent the closed-loop system as follows.

$$d\tilde{x}_t = \tilde{A}\tilde{x}_t dt + \sum_{i=1}^p \tilde{A}_i \tilde{x}_t dv_{it} + \tilde{G} d\beta_t \quad (3.7a)$$

$$= \tilde{A}\tilde{x}_t dt + [\tilde{A}_1 \tilde{x}_t \quad \tilde{A}_2 \tilde{x}_t \quad \dots \quad \tilde{A}_p \tilde{x}_t \quad \tilde{G}] \begin{bmatrix} dv_t \\ d\beta_t \end{bmatrix} \quad (3.7b)$$

In equation (3.7) we have defined the vector Wiener process

$$dv_t \triangleq [dv_{1t} \quad dv_{2t} \quad \dots \quad dv_{pt}]^T \quad (3.8)$$

and the closed-loop quantities, denoted by tildas, as follows:

$$\begin{aligned} \tilde{n} &\triangleq n + n_c, & \tilde{x}_t &\triangleq \begin{bmatrix} x_t \\ x_{ct} \end{bmatrix}, & \tilde{G} &\triangleq \begin{bmatrix} G_1 \\ B_c G_2 \end{bmatrix}, \\ \tilde{A} &\triangleq \begin{bmatrix} A & BC_c \\ B_c C & A_c \end{bmatrix}, & \tilde{A}_i &\triangleq \begin{bmatrix} A_i & B_i C_c \\ B_c C_i & 0 \end{bmatrix} \end{aligned} \quad (3.9)$$

Now the compensator design objective may be stated in terms of an LQ optimization problem. Define the standard cost functional,

$$J(A_c, B_c, C_c) \triangleq \lim_{T \rightarrow \infty} \frac{1}{T} E \left\{ \int_{t_0}^T [x_t^T R_1 x_t + 2x_t^T R_{12} u_t + u_t^T R_2 u_t] dt \right\} \quad (3.10)$$

In order to simplify the problem, the cost functional will be rewritten in an equivalent form without the integral,

$$J(A_c, B_c, C_c) = \lim_{t \rightarrow \infty} E \{ x_t^T R_1 x_t + 2x_t^T R_{12} u_t + u_t^T R_2 u_t \} \quad (3.11)$$

[Kwakernaak and Sivan 1972, p. 394, Theorem 5.4]. In fact, the limit may be eliminated as well, due to the assumed stationarity of the exogenous noise processes, provided the initial time is infinitely far in the past (i.e., for $t_0 \rightarrow -\infty$).

In light of the performance objective (3.11), the LQ optimal compensator design problem for systems with multiplicative white noise may be stated as follows. Given a plant (1.1)-(1.2) with p independent auxiliary inputs, find a matrix triple, $\{A_c, B_c, C_c\}$, to realize a compensator (3.5)-(3.6) which minimizes the cost functional (3.11).

3.2 Preliminaries of Stochastic Differential Equation Theory

Before deriving the necessary conditions for the optimization problem just stated, a brief overview will be given of those elements of stochastic differential equation theory which will be used in the derivation. For more information on matters where specific references are not given, see Jazwinski (1970) and references therein.

Consider the stochastic differential equation,

$$dx_t = f(x_t, t)dt + G(x_t, t)d\beta_t, \quad t \geq t_0 \quad (3.12)$$

where β_t is a vector Wiener process with $E\{d\beta_t d\beta_t^T\} = Q(t)dt$. Associated with this equation is the integral equation,

$$x_t - x_{t_0} = \int_{t_0}^t f(x_\tau, \tau)d\tau + \int_{t_0}^t G(x_\tau, \tau)d\beta_\tau \quad (3.13)$$

The first integral in (3.13) can be defined as a Riemann integral for sample functions of x , or as a *mean square Riemann integral* (Jazwinski 1970, p. 66) for x a stochastic process. If G were a function of time only, as would be the case for ordinary state equations with *additive* white noise, the second integral would be known as a *Wiener integral*. However, added complications arise when *multiplicative* white noise is present (G a function of x). In this more general case, the second integral may also be defined in a mean square sense, but such a definition will not be unique. Itô (1944) was the first to define this type of stochastic integral. In doing so he modeled the Riemann sum after a *forward difference equation*, effectively sampling the integrand at the beginning of each partition, and then proceeded with a mean square limit to define the integral. When the integral is interpreted in this sense, it is called an *Itô stochastic integral*, and the associated differential equation (3.12) is called an *Itô stochastic differential equation*, or *Itô equation*. The rules of calculus which result from this interpretation are called *Itô stochastic calculus*. Note that the closed-loop state equation, (3.7), can be interpreted as an Itô equation.

What is not immediately apparent is that the second integral in (3.13) has different interpretations according to where the partitions of the Riemann sum are sampled. For example, another important interpretation, called the *Stratonovich stochastic integral* (Stratonovich 1966), arises when the differential equation is modeled

as a *central difference equation* before taking the limit (i.e., partitions sampled in the center). Actually, the differences between the Itô and Stratonovich stochastic integrals are somewhat more profound than that, but the mathematical details which distinguish the two are beyond the scope of this work. The purpose of the foregoing discussion is to inform the reader that the stochastic differential equation (3.12) which appears in state-space models with multiplicative white noise has no meaning without assigning the associated integrals a particular interpretation (e.g., Itô or Stratonovich integrals).

The Itô stochastic integral is defined over a much broader class of functions than the Stratonovich integral and is used for most theoretical work in stability and control. Therefore, Itô stochastic calculus will be applied in the development of the next section. The Stratonovich integral does have a number of nice properties, but a simple transformation exists between differential equations of the two different interpretations (Jazwinski 1970, pp. 118–20, 131). It is not clear which interpretation is more “correct” for the application studied in this work. However, Wong and Zakai (1965) discovered a property of the Stratonovich noise model that gives it intuitive appeal for applications to physical systems:

Theorem 3.1: Let x_t be the solution to the stochastic differential equation (3.12). Now, replace the Wiener process β_t with a sequence of continuous piecewise linear approximations, $\beta_t^{(n)}$, such that $\beta_t^{(n)}$ converges to β_t as $n \rightarrow \infty$. Then the solutions, $x_t^{(n)}$, to the resulting sequence of ordinary differential equations converge to x_t if β_t in (3.12) is interpreted in the Stratonovich sense. This property does not hold in general for β_t interpreted in the Itô sense.

Because of Theorem 3.1 and the fact that Bernstein (e.g., 1987, Bernstein and Hyland 1988b) argues in favor of the Stratonovich interpretation for applications to lightly damped flexible structures, a comparison will be made in Chapter 6 of the performance of the two different controllers that result from the two stochastic integral interpretations when applied to a particular example.

An important result of Itô stochastic calculus, called *Itô's integration formula*, will be used in the next section. It is derived directly from *Itô's Theorem* below.

Theorem 3.2 (Itô's Theorem or Itô's Chain Rule) (e.g., Jazwinski 1970, p. 112, Lemma 4.2) Also, for a more formal proof, see Gikhman and Skorokhod (1969, pp. 387–91):

Let $\phi(x, t)$ be a real, scalar linear functional with continuous partial derivatives,

$$\phi_t \triangleq \frac{\partial \phi}{\partial t}, \quad \phi_x \triangleq \frac{\partial \phi}{\partial x}, \quad \phi_{xx} \triangleq \frac{\partial^2 \phi}{\partial x^2} \quad (3.14)$$

where x_t is the unique solution to the vector Itô stochastic differential equation, (3.12), and β_t is a vector Wiener process with $E\{d\beta_t d\beta_t^T\} = Q(t)dt$. Then the stochastic differential $d\phi$ of ϕ is

$$d\phi = \phi_t dt + \phi_x^T dx_t + \frac{1}{2} \text{trace}\{GQG^T \phi_{xx}\} dt \quad (3.15)$$

The existence of the third term in (3.15) establishes the fact that the usual chain rule does not hold in Itô stochastic calculus, although it does hold, incidentally, in Stratonovich stochastic calculus.

Integrating (3.15) and taking the expected value conditioned on $x_{t_0} = x_0$, Itô's integration formula results (Wonham 1970, p. 137):

$$\begin{aligned} E\{\phi(x_t, t) | x_{t_0} = x_0\} \\ = \phi(x_0, t_0) + E\left\{\int_{t_0}^t \left[\phi_\tau + \phi_x^T f + \frac{1}{2} \text{trace}\{GQG^T \phi_{xx}\}\right] d\tau | x_{t_0} = x_0\right\} \end{aligned} \quad (3.16)$$

The arguments, (x_τ, τ) , have been omitted for functions in the integrand in order to conserve space. The term in (3.15) involving $d\beta_t$ (by way of dx_t) does not appear in (3.16) due to the Martingale property of a stochastic integral w.r.t. a Wiener process. With this formula, we are now ready to find an expression for \tilde{Q}_t .

3.3 Conversion to Deterministic Minimization Problem

The first step in deriving the necessary conditions will be to simplify the optimization problem stated at the end of section 3.1 by converting it into a *deterministic* minimization problem. Define the closed-loop state covariance,

$$\tilde{Q}_t \triangleq E\{\tilde{x}_t \tilde{x}_t^T\}, \quad \tilde{Q} \triangleq \lim_{t \rightarrow \infty} \tilde{Q}_t \quad (3.17)$$

and the closed-loop state weighting matrix,

$$\tilde{R} \triangleq \begin{bmatrix} R_1 & R_{12}C_c \\ C_c^T R_{12}^T & C_c^T R_2 C_c \end{bmatrix} \quad (3.18)$$

Then the cost functional of (3.11) becomes,

$$J(A_c, B_c, C_c) = \lim_{t \rightarrow \infty} E\{\tilde{x}_t^T \tilde{R} \tilde{x}_t\} \quad (3.19a)$$

$$= \text{trace}\{\tilde{Q} \tilde{R}\} \quad (3.19b)$$

The objective then is to minimize the cost as described by (3.19b). The quantity \tilde{R} is known, because it is specified by the cost criterion in the problem statement. It remains, however, to find an expression for the closed-loop state covariance, \tilde{Q} .

Define the functional, $\phi_{ij}(\tilde{x}_t) = \tilde{x}_{ti} \tilde{x}_{tj}$, where \tilde{x}_{ti} denotes the i^{th} element of the solution vector \tilde{x}_t to the closed-loop state equation (3.7). Note that $\tilde{Q}_{ij}(i, j) = E\{\phi_{ij}(\tilde{x}_t)\}$. The f , G , and Q quantities of (3.7) as defined by the generic Itô equation, (3.12), are given by,

$$f(\tilde{x}_t) = \tilde{A} \tilde{x}_t, \quad G(\tilde{x}_t) = [\tilde{A}_1 \tilde{x}_t \quad \tilde{A}_2 \tilde{x}_t \quad \dots \quad \tilde{A}_p \tilde{x}_t \quad \tilde{G}]$$

$$Q = \text{diag}\{\gamma_1, \gamma_2, \dots, \gamma_p, V\} \quad (3.20)$$

The partial derivatives needed for Itô's integration formula are computed as follows.

$$\phi_t = 0, \quad (3.21)$$

$$\phi_{\tilde{x}}^T = \begin{cases} [0 \dots 0 \quad \tilde{x}_{tj} \quad 0 \dots 0 \quad \tilde{x}_{ti} \quad 0 \dots 0], & i \neq j \\ [0 \dots 0 \quad 2\tilde{x}_{ti} \quad 0 \dots 0], & i = j \end{cases} \quad (3.22a)$$

$$= \tilde{x}^T (E_{ij} + E_{ji}) \quad (3.22b)$$

where E_{ij} denotes the elementary matrix with 1 in the $(i, j)^{\text{th}}$ position and 0's elsewhere.

$$\phi_{\tilde{x}} \tilde{x} = \begin{cases} \begin{matrix} (i) & (j) \\ \begin{bmatrix} 0 & \dots & 0 \\ \vdots & \ddots & 1 & \vdots \\ \vdots & 1 & \ddots & \vdots \\ 0 & \dots & 0 \end{bmatrix} & i \neq j \\ \begin{matrix} (i) \\ \begin{bmatrix} 0 & \dots & 0 \\ \vdots & 2 & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \dots & 0 \end{bmatrix} & i = j \end{matrix} \end{cases} \quad (3.23a)$$

$$(i)$$

$$= (E_{ij} + E_{ji}) \quad (3.23b)$$

Therefore, the second and third terms of the integrand of Itô's integration formula are

$$\begin{aligned} \phi_{\tilde{x}}^T f &= \tilde{x}_{ti} (\tilde{A} \tilde{x}_t)_j + \tilde{x}_{tj} (\tilde{A} \tilde{x}_t)_i \\ &= (\tilde{A} \tilde{x}_t \tilde{x}_t^T)_{ji} + (\tilde{A} \tilde{x}_t \tilde{x}_t^T)_{ij} \\ &= (\tilde{x}_t \tilde{x}_t^T \tilde{A}^T)_{ij} + (\tilde{A} \tilde{x}_t \tilde{x}_t^T)_{ij} \end{aligned} \quad (3.24)$$

and

$$\begin{aligned} \text{trace}\{GQG^T \phi_{\tilde{x}} \tilde{x}\} &= \text{trace}\left\{ \begin{bmatrix} \tilde{A}_1 \tilde{x}_t & \tilde{A}_2 \tilde{x}_t & \dots & \tilde{A}_p \tilde{x}_t & \tilde{G} \end{bmatrix} \text{diag}\{\gamma_1, \gamma_2, \dots, \gamma_p, V\} \right. \\ &\quad \left. \begin{bmatrix} \tilde{A}_1 \tilde{x}_t & \tilde{A}_2 \tilde{x}_t & \dots & \tilde{A}_p \tilde{x}_t & \tilde{G} \end{bmatrix}^T (E_{ij} + E_{ji}) \right\} \\ &= \text{trace}\left\{ \left[\sum_{k=1}^p \gamma_k \tilde{A}_k \tilde{x}_t \tilde{x}_t^T \tilde{A}_k^T + \tilde{G} V \tilde{G}^T \right] (E_{ij} + E_{ji}) \right\} \\ &= 2 \left[\sum_{k=1}^p (\gamma_k \tilde{A}_k \tilde{x}_t \tilde{x}_t^T \tilde{A}_k^T)_{ij} + (\tilde{V})_{ij} \right] \end{aligned} \quad (3.25)$$

where $\tilde{V} \triangleq \tilde{G} V \tilde{G}^T$. In the last step of (3.25), the identity

$$\text{trace}\{A_{n \times m} B_{m \times n}\} = \sum_{i=1}^n \sum_{j=1}^m A_{ij} B_{ji} \quad (3.26)$$

was used, along with the symmetry of the expression enclosed in brackets.

Now we may apply Itô's integration formula to find an expression for the closed-loop state covariance, \tilde{Q}_t .

$$\begin{aligned} E\{\tilde{x}_t, \tilde{x}_t | \tilde{x}_{t_0} = \tilde{x}_0\} &= \tilde{x}_{t_0} \tilde{x}_{t_0}^T + E\left\{ \int_{t_0}^t [(\tilde{x}_\tau \tilde{x}_\tau^T \tilde{A}^T)_{ij} + (\tilde{A} \tilde{x}_\tau \tilde{x}_\tau^T)_{ij} \right. \\ &\quad \left. + \sum_{k=1}^p (\gamma_k \tilde{A}_k \tilde{x}_\tau \tilde{x}_\tau^T \tilde{A}_k^T)_{ij} + (\tilde{V})_{ij}] d\tau \mid \tilde{x}_{t_0} = \tilde{x}_0 \right\} \end{aligned} \quad (3.27)$$

$$\begin{aligned} \tilde{Q}_t(i, j) &= \tilde{Q}_{t_0}(i, j) + \int_{t_0}^t [(\tilde{Q}_\tau \tilde{A}^T)_{ij} + (\tilde{A} \tilde{Q}_\tau)_{ij} \\ &\quad + \sum_{k=1}^p (\gamma_k \tilde{A}_k \tilde{Q}_\tau \tilde{A}_k^T)_{ij} + (\tilde{V})_{ij}] d\tau \end{aligned} \quad (3.28)$$

$$\tilde{Q}_t = \tilde{Q}_{t_0} + \int_{t_0}^t [\tilde{Q}_\tau \tilde{A}^T + \tilde{A} \tilde{Q}_\tau + \sum_{i=1}^p \gamma_i \tilde{A}_i \tilde{Q}_\tau \tilde{A}_i^T + \tilde{V}] d\tau \quad (3.29)$$

By the Fundamental Theorem of Integral Calculus,

$$\dot{\tilde{Q}}_t = \tilde{A} \tilde{Q}_t + \tilde{Q}_t \tilde{A}^T + \sum_{i=1}^p \gamma_i \tilde{A}_i \tilde{Q}_t \tilde{A}_i^T + \tilde{V} \quad (3.30)$$

The cost functional, (3.19b), requires the steady state solution to (3.30), but first we must determine under what conditions such a solution exists. To that end, we will consider an alternate expression for \tilde{Q}_t which is useful for examining the *stochastic stability* of the system. The following lemma will be used repeatedly.

Lemma 3.1: Given matrices A , B , and C of compatible dimensions,

$$\text{vec}(ABC) = (C^T \otimes A) \text{vec}(B) \quad (3.31)$$

Proof: See Graham (1981, p. 25, Property VIII).

In Lemma 3.1, the "vec" operator merely organizes the columns of its matrix argument into a single column vector with the first column on top and each subsequent column

beneath the previous one. The symbol " \otimes " represents the Kronecker product. Now define

$$\mathcal{A} \triangleq \tilde{A} \oplus \tilde{A} + \sum_{i=1}^p \gamma_i \tilde{A}_i \otimes \tilde{A}_i \quad (3.32)$$

where " \oplus " is the Kronecker sum. Then taking the "vec" operation of (3.30) and applying Lemma 3.1 to the first three terms on the right hand side, we obtain the alternate expression,

$$\text{vec}(\dot{\tilde{Q}}) = \mathcal{A} \text{vec}(\tilde{Q}) + \text{vec}(\tilde{V}) \quad (3.33)$$

From this equation we see that when \mathcal{A} is a stable matrix, a steady state solution, \tilde{Q} , exists. Under this condition (i.e., stable \mathcal{A}) the closed-loop system (3.7) is said to be *second-moment stable* (Arnold 1974), and \tilde{Q} is the solution to the algebraic equation,

$$0 = \tilde{A}\tilde{Q} + \tilde{Q}\tilde{A}^T + \sum_{i=1}^p \gamma_i \tilde{A}_i \tilde{Q} \tilde{A}_i^T + \tilde{V} \quad (3.34)$$

which is a modified Lyapunov equation. Note that when the plant uncertainties are zero (i.e., $\gamma_i = 0$ or $\tilde{A}_i = 0$, for all $i = 1, \dots, p$) the third term in (3.34) vanishes and the standard Lyapunov equation results for the steady-state second-moment matrix of the state vector for a system driven by white noise [see e.g., Kwakernaak and Sivan (1972, p.101, Theorem 1.52)]. In order to assure that the cost functional, J , is finite and independent of the system initial values, only second-moment stabilizing compensators will be admissible.

Therefore the original stochastic optimization problem has been reduced to the following deterministic one:

| | |
|---|--------|
| <p>minimize: $\text{trace}(\tilde{Q}\tilde{R})$</p> <p>over the set: $\{(A_c, B_c, C_c): \mathcal{A} \text{ is stable}\}$</p> <p>subject to: $0 = \tilde{A}\tilde{Q} + \tilde{Q}\tilde{A}^T + \sum_{i=1}^p \gamma_i \tilde{A}_i \tilde{Q} \tilde{A}_i^T + \tilde{V}$</p> | (3.35) |
|---|--------|

3.4 Derivation of the First-Order Necessary Conditions

The first-order necessary conditions that a controller triple, $\{A_c, B_c, C_c\}$, must meet to solve this type of problem have been derived by at least two different methods — the Lagrange multiplier method (Bernstein and Haddad, 1989), used to solve an \mathcal{H}_∞ problem, and the calculus of variations method (Bernstein and Hyland, 1988a). The Lagrange multiplier solution is somewhat easier to follow but could not be found in complete form in the literature for this particular problem. Therefore, the bulk of it will be derived here, leading up to the simple but more tedious algebraic manipulations, which will be referenced to a paper where they are found in their entirety.

The Lagrangian corresponding to the optimization problem (3.35), can be written as,

$$\mathcal{L}(A_c, B_c, C_c, \tilde{Q}, \tilde{P}, \lambda) = \text{tr}\{\lambda \tilde{Q} \tilde{R} + [\tilde{A} \tilde{Q} + \tilde{Q} \tilde{A}^T + \sum_{i=1}^p \gamma_i \tilde{A}_i \tilde{Q} \tilde{A}_i^T + \tilde{V}] \tilde{P}\} \quad (3.36)$$

where \tilde{P} is an $\tilde{n} \times \tilde{n}$ matrix of Lagrange multipliers and λ is the scalar supplementary Lagrange multiplier, which is without loss of generality equal to 1 if the problem is *normal* (i.e., $\lambda = 0 \Rightarrow \tilde{P} = 0$) — see, for example, Ewing (1985, sec. 5.5). Since the conventional Lagrange multiplier problem involves a *vector* of constraints and hence a *vector* of Lagrange multipliers, the use of a Lagrange multiplier *matrix* requires some justification. The conventional problem with equality constraints takes the form,

$$\begin{aligned} &\text{minimize: } f(x, y) \\ &\text{subject to: } g(x) = 0 \end{aligned} \quad (3.37)$$

where f is a scalar function of the vector variables x and y , and g is a vector function of x . Then the Lagrangian to be minimized becomes,

$$\mathcal{L}(x, y, p, \lambda) = \lambda f + p^T g \quad (3.38)$$

with p a column vector of Lagrange multipliers with dimensions equal to those of x . The matrix equality constraint in (3.35) may be rewritten in the form (3.38) by applying the “vec” operator to the equation. Define the matrix expression on the right hand side of the equality constraint in (3.35) as X . Then an equivalent to the constraint $X = 0$ is $\text{vec}(X) = 0$, which yields the Lagrangian,

$$\mathcal{L}(A_c, B_c, C_c, \tilde{Q}, p, \lambda) = \lambda \operatorname{tr}\{\tilde{Q}\tilde{R}\} + p^T \operatorname{vec}(X) \quad (3.39)$$

where p represents an $\tilde{n}^2 \times 1$ Lagrange multiplier vector. Of course, (3.39) is equivalent to (3.36). Applying the identity, $(\operatorname{vec} A^T)^T \operatorname{vec}(B) = \operatorname{tr}(AB)$ (Graham 1981, p. 18, Example 1.4), to the term involving p , we see that (3.39) may be written as

$$\mathcal{L}(A_c, B_c, C_c, \tilde{Q}, \tilde{P}, \lambda) = \lambda \operatorname{tr}\{\tilde{Q}\tilde{R}\} + \operatorname{tr}\{\tilde{P}^T X\} \quad (3.40)$$

where the $\tilde{n} \times \tilde{n}$ Lagrange multiplier matrix \tilde{P} is defined by $p = \operatorname{vec}(\tilde{P})$. Clearly, (3.40) is equivalent to (3.36). Note that the symmetry of the expression defined as X means that we may assume without loss of generality that \tilde{P} is symmetric.

In taking the partial derivatives of the Lagrangian, the following properties will be used (Graham 1981, pp. 76-78, Examples 5.4-5.6),

$$\frac{\partial \operatorname{tr}(AX)}{\partial X} = A^T, \quad \frac{\partial \operatorname{tr}(AX^T)}{\partial X} = A, \quad \frac{\partial \operatorname{tr}(X^T A X B)}{\partial X} = A X B + A^T X B^T \quad (3.41)$$

Taking the partial derivatives of the Lagrangian, (3.36), with respect to each of its arguments and setting each equal to zero:

$$\boxed{1} \quad \frac{\partial \mathcal{L}}{\partial \tilde{Q}} = 0 :$$

$$\lambda \tilde{R} + \tilde{A}^T \tilde{P} + \tilde{P} \tilde{A} + \sum_{i=1}^p \gamma_i \tilde{A}_i^T \tilde{P} \tilde{A}_i = 0 \quad (3.42)$$

In order to test whether the optimization problem is normal, we set $\lambda = 0$ in (3.42). Taking the "vec" operation of the equation, as well, the result is

$$\left(\tilde{A}^T \oplus \tilde{A}^T + \sum_{i=1}^p \gamma_i \tilde{A}_i^T \otimes \tilde{A}_i^T \right) \operatorname{vec} \tilde{P} = 0 \quad (3.43)$$

$$\mathcal{A}^T \operatorname{vec} \tilde{P} = 0$$

Since \mathcal{A} is stable by assumption, \mathcal{A}^T has no nullspace, and therefore $\tilde{P} = 0$. As stated above, $\lambda = 0 \Rightarrow \tilde{P} = 0$ means that the problem is normal, and we can take $\lambda = 1$, without loss of generality. Therefore, with $\lambda = 1$,

$$\boxed{\tilde{R} + \tilde{A}^T \tilde{P} + \tilde{P} \tilde{A} + \sum_{i=1}^p \gamma_i \tilde{A}_i^T \tilde{P} \tilde{A}_i = 0} \quad (3.44)$$

Before proceeding, partition the symmetric matrices, \tilde{V} , \tilde{Q} , and \tilde{P} , into 2×2 blocks (of dimensions $n \times n$, $n \times n_c$, $n_c \times n$, and $n_c \times n_c$):

$$\tilde{V} \triangleq \begin{bmatrix} V_1 & V_{12} B_c^T \\ B_c V_{12}^T & B_c V_2 B_c^T \end{bmatrix}, \quad \tilde{Q} \triangleq \begin{bmatrix} Q_1 & Q_{12} \\ Q_{12}^T & Q_2 \end{bmatrix}, \quad \tilde{P} \triangleq \begin{bmatrix} P_1 & P_{12} \\ P_{12}^T & P_2 \end{bmatrix} \quad (3.45)$$

Now the Lagrangian may be expanded:

$$\mathcal{L}(A_c, B_c, C_c, \tilde{Q}, \tilde{P}) = \text{tr}\{Q_1 R_1 + Q_{12} C_c^T R_{12}^T + Q_{12}^T R_{12} C_c + Q_2 C_c^T R_2 C_c \quad (3.46)$$

$$+ (A Q_1 + B C_c Q_{12}^T) P_1 + (A Q_{12} + B C_c Q_2) P_{12}^T \\ + (B_c C Q_1 + A_c Q_{12}^T) P_{12} + (B_c C Q_{12} + A_c Q_2) P_2$$

$$+ (Q_1 A^T + Q_{12} C_c^T B^T) P_1 + (Q_1 C^T B_c^T + Q_{12} A_c^T) P_{12}^T \\ + (Q_{12}^T A^T + Q_2 C_c^T B^T) P_{12} + (Q_{12}^T C^T B_c^T + Q_2 A_c^T) P_2$$

$$+ \sum_{i=1}^p \gamma_i [(A_i Q_1 A_i^T + B_i C_c Q_{12}^T A_i^T + A_i Q_{12} C_c^T B_i^T \\ + B_i C_c Q_2 C_c^T B_i^T) P_1 \\ + (A_i Q_1 C_i^T B_c^T + B_i C_c Q_{12}^T C_i^T B_c^T) P_{12}^T \\ + (B_c C_i Q_1 A_i^T + B_c C_i Q_{12} C_c^T B_i^T) P_{12} \\ + (B_c C_i Q_1 C_i^T B_c^T) P_2]$$

$$+ V_1 P_1 + V_{12} B_c^T P_{12}^T + B_c V_{12}^T P_{12} + B_c V_2 B_c^T P_2\}$$

This allows us to take the partial derivatives of the Lagrangian with respect to the compensator parameters.

$$\textcircled{2} \quad \frac{\partial \mathcal{L}}{\partial A_c} = 0 :$$

$$P_{12}^T Q_{12} + P_2 Q_2 + P_{12}^T Q_{12} + P_2 Q_2 = 0 \quad (3.47)$$

$$\boxed{P_{12}^T Q_{12} + P_2 Q_2 = 0} \quad (3.48)$$

$$\textcircled{3} \quad \frac{\partial \mathcal{L}}{\partial B_c} = 0 :$$

$$P_{12}^T Q_1 C^T + P_2 Q_{12}^T C^T + P_{12}^T Q_1 C^T + P_2 Q_{12}^T C^T \quad (3.49)$$

$$+ \sum_{i=1}^p \gamma_i (P_{12}^T A_i Q_1 C_i^T + P_{12}^T B_i C_c Q_{12}^T C_i^T + P_{12}^T A_i Q_1 C_i^T + P_{12}^T B_i C_c Q_{12}^T C_i^T + 2P_2 B_c C_i Q_1 C_i^T)$$

$$+ P_{12}^T V_{12} + P_{12}^T V_{12} + 2P_2 B_c V_2 = 0$$

$$(P_{12}^T Q_1 + P_2 Q_{12}^T) C^T + P_{12}^T V_{12} + P_2 B_c V_2 \quad (3.50)$$

$$+ \sum_{i=1}^p \gamma_i (P_{12}^T A_i Q_1 + P_{12}^T B_i C_c Q_{12}^T + P_2 B_c C_i Q_1) C_i^T = 0$$

If for each i , B_i and C_i are not both nonzero (i.e., the Wiener processes used for control and measurement dependent noise are independent), a closed form expression for B_c can be found. For then we have,

$$(P_{12}^T Q_1 + P_2 Q_{12}^T) C^T + P_{12}^T V_{12} + P_2 B_c V_2 \quad (3.51)$$

$$+ \sum_{i=1}^p \gamma_i (P_{12}^T A_i Q_1 + P_2 B_c C_i Q_1) C_i^T = 0$$

which leads to,

$$B_c = -P_2^{-1} [(P_{12}^T Q_1 + P_2 Q_{12}^T) C^T + P_{12}^T (V_{12} + \sum_{i=1}^p \gamma_i A_i Q_1 C_i^T)] \hat{V}_2^{-1} \quad (3.52)$$

where

$$\hat{V}_2 \triangleq V_2 + \sum_{i=1}^p \gamma_i C_i Q_1 C_i^T \quad (3.53)$$

$$\boxed{4} \quad \frac{\partial \mathcal{L}}{\partial C_c} = 0 :$$

$$R_{12}^T Q_{12} + R_{12}^T Q_{12} + 2R_2 C_c Q_2 \quad (3.54)$$

$$\begin{aligned} & + B^T P_1 Q_{12} + B^T P_{12} Q_2 + B^T P_1 Q_{12} + B^T P_{12} Q_2 \\ & + \sum_{i=1}^p \gamma_i (B_i^T P_1 A_i Q_{12} + B_i^T P_1 A_i Q_{12} + 2B_i^T P_1 B_i C_c Q_2 \\ & + B_i^T P_{12} B_i C_c Q_{12} + B_i^T P_{12} B_i C_c Q_{12}) = 0 \end{aligned}$$

$$R_{12}^T Q_{12} + R_2 C_c Q_2 + B^T (P_1 Q_{12} + P_{12} Q_2) \quad (3.55)$$

$$+ \sum_{i=1}^p \gamma_i B_i^T (P_1 A_i Q_{12} + P_1 B_i C_c Q_2 + P_{12} B_i C_c Q_{12}) = 0$$

Assuming as before that for each i , B_i and C_i are not both nonzero,

$$\boxed{\begin{aligned} & R_{12}^T Q_{12} + R_2 C_c Q_2 + B^T (P_1 Q_{12} + P_{12} Q_2) \\ & + \sum_{i=1}^p \gamma_i B_i^T (P_1 A_i Q_{12} + P_1 B_i C_c Q_2) = 0 \end{aligned}} \quad (3.56)$$

Then we find that

$$C_c = -\hat{R}_2^{-1} [B^T (P_1 Q_{12} + P_{12} Q_2) + (R_{12}^T + \sum_{i=1}^p \gamma_i B_i^T P_1 A_i) Q_{12}] Q_2^{-1} \quad (3.57)$$

where

$$\hat{R}_2 \triangleq R_2 + \sum_{i=1}^p \gamma_i B_i^T P_1 B_i \quad (3.58)$$

The equality constraint in (3.35) and the framed equations — (3.44), (3.48), (3.51), and (3.56) — are equivalent to equations (20), (71), (74), (75), and (76), respectively, in (Bernstein and Hyland 1988a). There, the derivation of the first-order

necessary conditions is completed. Essentially, it involves: (1) proving the existence of the inverses in (3.52) and (3.57); (2) expanding both (3.44) and the equality constraint in (3.35) into four blocks; (3) substituting the expressions for B_c and C_c into those expanded equations; and (4) using algebraic manipulations to solve for an expression for A_c and to find equations for the unknowns, \tilde{Q} and \tilde{R} , independent of A_c , B_c , and C_c . The necessary conditions in final form are:

$$0 = AQ + QA^T - Q\hat{V}_2^{-1}Q^T + V_1 \quad (3.59)$$

$$+ \sum_{i=1}^p \gamma_i [A_i Q A_i^T + (A_i - B_i \hat{R}_2^{-1} \mathcal{P}) \hat{Q} (A_i - B_i \hat{R}_2^{-1} \mathcal{P})^T]$$

$$0 = A^T P + PA - \mathcal{P}^T \hat{R}_2^{-1} \mathcal{P} + R_1 \quad (3.60)$$

$$+ \sum_{i=1}^p \gamma_i [A_i^T P A_i + (A_i - Q \hat{V}_2^{-1} C_i)^T \hat{P} (A_i - Q \hat{V}_2^{-1} C_i)]$$

$$0 = A_P \hat{Q} + \hat{Q} A_P^T + Q \hat{V}_2^{-1} Q^T \quad (3.61)$$

$$0 = A_Q^T \hat{P} + \hat{P} A_Q + \mathcal{P}^T \hat{R}_2^{-1} \mathcal{P} \quad (3.62)$$

where

$$Q \triangleq QC^T + V_{12} + \sum_{i=1}^p \gamma_i A_i (Q + \hat{Q}) C_i^T \quad (3.63)$$

$$\mathcal{P} \triangleq B^T P + R_{12}^T + \sum_{i=1}^p \gamma_i B_i^T (P + \hat{P}) A_i \quad (3.64)$$

$$A_Q \triangleq A - Q \hat{V}_2^{-1} C, \quad A_P \triangleq A - B \hat{R}_2^{-1} \mathcal{P} \quad (3.65)$$

and A_c , B_c , and C_c are written in terms of the unknown variables Q , P , \hat{Q} , and \hat{P} as:

$$A_c = A - B\hat{R}_2^{-1}\mathcal{P} - \mathcal{Q}\hat{V}_2^{-1}C \quad (3.66)$$

$$B_c = \mathcal{Q}\hat{V}_2^{-1} \quad (3.67)$$

$$C_c = -\hat{R}_2^{-1}\mathcal{P} \quad (3.68)$$

3.5 Modifications for Stratonovich Noise Model

As mentioned in section 3.2, there is a simple transformation which allows us to reinterpret the Itô equation (3.7b) — that is, our state equation — in the sense of Stratonovich. That transformation is given by the following theorem.

Theorem 3.3: Given the vector Stratonovich equation,

$$dx_t = f(x_t, t)dt + G(x_t, t)d\beta_t, \quad t \geq t_0 \quad (3.69)$$

where x_t has dimensions $n \times 1$ and β_t is a vector Wiener process with $E\{d\beta_t d\beta_t^T\} = Q(t)dt$, the equivalent Itô equation for the i^{th} element of dx_t is

$$dx_{it} = \left\{ f_i(x_t, t) + \frac{1}{2} \sum_{k=1}^n \left[G(x_t, t) V \frac{\partial G(x_t, t)}{\partial x_{it}} \right]_{ki} \right\} dt + [G(x_t, t)d\beta_t]_i, \quad t \geq t_0 \quad (3.70)$$

This theorem is given in Jazwinski (1970, p. 131) and is formally proved by Stratonovich (1966). It shows that a correction term must be added to the first term in the state equations. Comparing (3.69) with (3.7b),

$$d\tilde{x}_t = \tilde{A}\tilde{x}_t dt + \begin{bmatrix} \tilde{A}_1 \tilde{x}_t & \tilde{A}_2 \tilde{x}_t & \dots & \tilde{A}_p \tilde{x}_t & \tilde{G} \end{bmatrix} \begin{bmatrix} dv_t \\ d\beta_t \end{bmatrix} \quad (3.7b)$$

and recalling equations (3.20),

$$f(\tilde{x}_t) = \tilde{A}\tilde{x}_t, \quad G(\tilde{x}_t) = \begin{bmatrix} \tilde{A}_1 \tilde{x}_t & \tilde{A}_2 \tilde{x}_t & \dots & \tilde{A}_p \tilde{x}_t & \tilde{G} \end{bmatrix}$$

$$Q = \text{diag}\{\gamma_1, \gamma_2, \dots, \gamma_p, V\} \quad (3.20)$$

we can substitute the quantities from (3.20) into (3.70) and show that in order to give the multiplicative white noise model a Stratonovich interpretation, we need to make the following substitution,

$$\tilde{A}\tilde{x}_i \leftarrow \tilde{A}\tilde{x}_i + \frac{1}{2} \sum_{i=1}^p \gamma_i \tilde{A}_i^2 \tilde{x}_i \quad (3.71)$$

Eliminating the common factor, \tilde{x}_i , and expanding \tilde{A} and \tilde{A}_i into 2×2 block form [see (3.9)], we have

$$\begin{bmatrix} A & BC_c \\ B_c C & A_c \end{bmatrix} \leftarrow \begin{bmatrix} A & BC_c \\ B_c C & A_c \end{bmatrix} + \frac{1}{2} \sum_{i=1}^p \gamma_i \begin{bmatrix} A_i^2 + B_i C_c B_c C_i & A_i B_i C_c \\ B_c C_i A_i & B_c C_i B_i C_c \end{bmatrix} \quad (3.72)$$

Then employing once again the assumption that for each i , B_i and C_i are not both nonzero, and examining the (1,1), (1,2), and (2,1) blocks of (3.72), we find that the Stratonovich noise interpretation may be accomplished by means of the three simple substitutions in the plant model,

$$\begin{array}{l} A \leftarrow A + \frac{1}{2} \sum_{i=1}^p \gamma_i A_i^2, \\ B \leftarrow B + \frac{1}{2} \sum_{i=1}^p \gamma_i A_i B_i, \\ C \leftarrow C + \frac{1}{2} \sum_{i=1}^p \gamma_i C_i A_i \end{array} \quad (3.73)$$

3.6 Controller Design Summary

The multiplicative white noise model of this chapter is used for the auxiliary inputs of the LQG/PRE error model. This auxiliary input model is combined with the standard cost functional penalties on the auxiliary outputs to produce a new parameter-robust design procedure. The performance/robustness tradeoff is accomplished similarly to LQG/PRE, where the noise intensities are adjusted by means of the scalar parameters γ_i ($i = 1, 2, \dots, p$), and auxiliary output penalties are adjusted as before, by

modifying the state and control weighting matrices (R_1 and R_2 , respectively) as in equation (1.9) and substituting these modified matrices into the controller design equations developed in this chapter.

4. Optimal Reduced-Order Control

One of the great advantages of the parameter robust design techniques discussed in Chapters 2 and 3 (as well as LQG/PRE) over μ -synthesis (see section 1.1B) is the fact that they lend themselves to application of a method of optimal reduced-order controller design, called the *optimal projection equations* (Hyland and Bernstein 1984). As is well-known, the LQG-optimal compensator for an n^{th} -order plant model is also of order n . However, the optimal projection equations allow us to specify a compensator order $n_c < n$ and directly design the optimal compensator of that order, provided a stabilizing n_c^{th} -order compensator exists. The optimal projection results will be stated here directly. For a proof, see Hyland and Bernstein (1984) [for the case of no cross-weighting in the cost and no cross-covariance between process and sensor noise], or see Bernstein and Hyland (1988a) for the more general case.

Given the n^{th} -order plant,

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) + G_1 w(t) \\ y(t) &= Cx(t) + G_2 w(t)\end{aligned}\tag{4.1}$$

with uncorrelated, unit intensity, Gaussian white noise vector, w , define the covariance matrices,

$$V_1 \triangleq G_1 G_1^T, \quad V_{12} \triangleq G_1 G_2^T, \quad V_2 \triangleq G_2 G_2^T\tag{4.2}$$

and the linear-quadratic cost functional,

$$J(A_c, B_c, C_c) = \lim_{T \rightarrow \infty} E\{x_t^T R_1 x_t + 2x_t^T R_{12} u_t + u_t^T R_2 u_t \, dt\}\tag{4.3}$$

If a stabilizing n_c^{th} -order compensator,

$$\begin{aligned}x_c(t) &= A_c x_c(t) + B_c y(t) \\ u(t) &= C_c x_c(t)\end{aligned}\tag{4.4}$$

exists, the one which minimizes (4.3) is given by the design equations,

$$A_c = \Gamma(A - BR_2^{-1}\mathfrak{P} - \mathcal{Q}V_2^{-1}C)G^T \quad (4.5)$$

$$B_c = \Gamma\mathcal{Q}V_2^{-1}$$

$$C_c = -R_2^{-1}\mathfrak{P}G^T$$

where

$$\mathcal{Q} \triangleq QC^T + V_{12}, \quad \mathfrak{P} \triangleq B^TP + R_{12}^T \quad (4.6)$$

Also define

$$A_Q \triangleq A - \mathcal{Q}V_2^{-1}C, \quad A_P \triangleq A - BR_2^{-1}\mathfrak{P} \quad (3.65)$$

Then the $n \times n$ matrices Q , P , \hat{Q} , \hat{P} , and τ_{\perp} are the solution to the optimal projection equations,

$$0 = AQ + QA^T - \mathcal{Q}V_2^{-1}\mathcal{Q}^T + V_1 + \tau_{\perp}\mathcal{Q}V_2^{-1}\mathcal{Q}^T\tau_{\perp}^T \quad (4.7)$$

$$0 = A^TP + PA - \mathfrak{P}^TR_2^{-1}\mathfrak{P} + R_1 + \tau_{\perp}^T\mathfrak{P}^TR_2^{-1}\mathfrak{P}\tau_{\perp} \quad (4.8)$$

$$0 = A_P\hat{Q} + \hat{Q}A_P^T + \mathcal{Q}V_2^{-1}\mathcal{Q}^T - \tau_{\perp}\mathcal{Q}V_2^{-1}\mathcal{Q}^T\tau_{\perp}^T \quad (4.9)$$

$$0 = A_Q^T\hat{P} + \hat{P}A_Q + \mathfrak{P}^TR_2^{-1}\mathfrak{P} - \tau_{\perp}^T\mathfrak{P}^TR_2^{-1}\mathfrak{P}\tau_{\perp} \quad (4.10)$$

$$\text{rank}(\hat{Q}) = \text{rank}(\hat{P}) = \text{rank}(\hat{Q}\hat{P}) = n_c \quad (4.11)$$

where $\tau_{\perp} \triangleq I - \tau$ and the optimal projection matrix, τ , is perhaps most simply described as follows. Define a *balancing transformation* Ψ (Laub 1980) that simultaneously diagonalizes \hat{Q} and \hat{P} by,

$$\Psi^{-1}\hat{Q}\Psi^{-1} = \Psi^T\hat{P}\Psi = \Lambda \quad (4.12)$$

$$\Lambda \triangleq \text{diag}\{\lambda_1, \lambda_2, \dots, \lambda_{n_c}, 0, \dots, 0\}$$

Then

$$\tau = \Psi \begin{bmatrix} I_{n_c} & 0 \\ 0 & 0 \end{bmatrix} \Psi^{-1} \quad (4.13)$$

Also,

$$G = [I_{n_c} \ 0] \Psi^T \quad \text{and} \quad \Gamma = [I_{n_c} \ 0] \Psi^{-1} \quad (4.14)$$

Note that the modified Riccati equations (4.7)–(4.8) and the modified Lyapunov equations (4.9)–(4.10) are coupled together by the matrix τ_{\perp} . For the case of a full-order compensator ($n_c = n$), we have $\tau = G = \Gamma = I$ and $\tau_{\perp} = 0$, so the terms involving τ_{\perp} disappear. In that case (4.7) and (4.8) become the standard observer and regulator Riccati equations of LQG control, and equations (4.5) become the standard state-space solution for the controller. Also, in the full-order case, the Lyapunov equations (4.9) and (4.10) become superfluous. Notably, these equations must be solved for the observability gramian, \hat{Q} , and the controllability gramian, \hat{P} , if one wishes to balance and truncate the plant to produce a suboptimal reduced-order controller. The coupled structure of the optimal projection equations shows that, in a sense, the balancing and controller design must be carried out simultaneously in order to preserve optimality. In the general case, where possibly $n_c < n$, \hat{Q} and \hat{P} are referred to in the literature as the observability and controllability *pseudogramians*.

Clearly, the optimal reduced-order design equations apply to LQG/PRE, since LQG/PRE differs from standard LQG only in the fact that the values of V_1 , V_2 , R_1 , and R_2 are modified, as in (1.9). Optimal projection may also easily be applied to the frequency-domain method of Chapter 2. The auxiliary input modeling phase of that method merely involves augmentation of the plant dynamics and augmentation of the covariance matrices V_1 and V_2 to account for additional noise sources. The auxiliary output modeling phase only modifies the weighting matrices R_1 and R_2 . Therefore, the modified model of the plant, the modified noise model, and the modified cost criterion still constitute a standard LQG problem. That being the case, the optimal projection equations may be applied directly.

The time-domain method of Chapter 3, however, does not conform to the standard LQG framework, because the multiplicative white noise model of the auxiliary

inputs requires a change in the structure of the state and/or output equations of the plant. Fortunately, optimal projection may still be applied. The optimal projection equations, as stated above, may be derived by the Lagrange multiplier method of section 3.4 (Hyland and Bernstein 1984). More specifically, the rank conditions (4.11) are enforced in the course of the algebraic manipulations which follow after taking the partial derivatives of the Lagrangian. In fact, the same rank conditions may be applied to the more general multiplicative white noise problem (Bernstein and Hyland 1988a). These rank conditions did not appear explicitly in Chapter 3, because the algebraic manipulations were referenced to the paper just mentioned. Also, Chapter 3 was concerned only with the case of a full-order controller. Of course, the full-order case is still important, since a restriction, $n_c < n$, on the controller order generally results in an increase in the value of the cost functional, J , and therefore brings about a degradation in performance.

By including multiplicative white noise in the state and output equations of the plant, as in Chapter 3,

$$dx_t = Ax_t dt + \sum_{i=1}^p \gamma_i A_i x_t dv_{it} + Bu_t dt + \sum_{i=1}^p \gamma_i B_i u_t dv_{it} + G_1 d\beta_t \quad (3.1)$$

$$dy_t = Cx_t dt + \sum_{i=1}^p \gamma_i C_i x_t dv_{it} + G_2 d\beta_t \quad (3.2)$$

we obtain a general model which applies to all of the parameter-robust controller design methods discussed in this work. For a method in which multiplicative white noise is not desired, we may set $A_i = B_i = C_i = 0$ in (3.1)-(3.2) to obtain (4.1). The optimal n_c^{th} -order compensator (for $n_c \leq n$), if a stabilizing compensator of that order exists, is given by

$$A_c = \Gamma(A - B\hat{R}_2^{-1}\mathcal{P} - Q\hat{V}_2^{-1}C)G^T \quad (4.15)$$

$$B_c = \Gamma Q\hat{V}_2^{-1}$$

$$C_c = -\hat{R}_2^{-1}\mathcal{P}G^T$$

where Q , P , \hat{Q} , \hat{P} , and τ_{\perp} are the solution to the more general optimal projection equations,

$$0 = AQ + QA^T - Q\hat{V}_2^{-1}Q^T + V_1 + \tau_{\perp} Q\hat{V}_2^{-1}Q^T \tau_{\perp}^T \quad (4.16)$$

$$+ \sum_{i=1}^p \gamma_i [A_i Q A_i^T + (A_i - B_i \hat{R}_2^{-1} \mathfrak{P}) \hat{Q} (A_i - B_i \hat{R}_2^{-1} \mathfrak{P})^T]$$

$$0 = A^T P + PA - \mathfrak{P}^T \hat{R}_2^{-1} \mathfrak{P} + R_1 + \tau_{\perp}^T \mathfrak{P}^T \hat{R}_2^{-1} \mathfrak{P} \tau_{\perp} \quad (4.17)$$

$$+ \sum_{i=1}^p \gamma_i [A_i^T P A_i + (A_i - Q\hat{V}_2^{-1} C_i)^T \hat{P} (A_i - Q\hat{V}_2^{-1} C_i)]$$

$$0 = A_P \hat{Q} + \hat{Q} A_P^T + Q\hat{V}_2^{-1}Q^T - \tau_{\perp} Q\hat{V}_2^{-1}Q^T \tau_{\perp}^T \quad (4.18)$$

$$0 = A_Q^T \hat{P} + \hat{P} A_Q + \mathfrak{P}^T \hat{R}_2^{-1} \mathfrak{P} - \tau_{\perp}^T \mathfrak{P}^T \hat{R}_2^{-1} \mathfrak{P} \tau_{\perp} \quad (4.19)$$

$$\text{rank}(\hat{Q}) = \text{rank}(\hat{P}) = \text{rank}(\hat{Q}\hat{P}) = n_c \quad (4.20)$$

The $n_c \times n$ matrices Γ and G , and the $n \times n$ matrix τ_{\perp} are determined by \hat{Q} and \hat{P} as described above, and the following definitions from Chapter 3 apply.

$$\hat{V}_2 \triangleq V_2 + \sum_{i=1}^p \gamma_i C_i Q_i C_i^T \quad (3.53)$$

$$\hat{R}_2 \triangleq R_2 + \sum_{i=1}^p \gamma_i B_i^T P_i B_i \quad (3.58)$$

$$Q \triangleq QC^T + V_{12} + \sum_{i=1}^p \gamma_i A_i (Q + \hat{Q}) C_i^T \quad (3.63)$$

$$\mathfrak{P} \triangleq B^T P + R_{12}^T + \sum_{i=1}^p \gamma_i B_i^T (P + \hat{P}) A_i \quad (3.64)$$

$$A_Q \triangleq A - Q\hat{V}_2^{-1}C, \quad A_P \triangleq A - B\hat{R}_2^{-1}\mathfrak{P} \quad (3.65)$$

The numerical solution of the coupled Riccati- and Lyapunov-type equations in this and the previous chapter require an iterative algorithm. The algorithms used to solve these equations will be discussed in Chapter 5.

5. Methods and Algorithms for Controller Design

Chapters 2, 3, and 4 described the basic theory needed for the controller design techniques under study. This chapter is concerned with the practical problems involved in applying the design methods to an actual control system. In the sections that follow, a few of the more important implementation considerations are discussed which arose over the course of this research. Solutions are given to these problems, and algorithms are described in detail.

5.1 Selection of Cost Functional Weighting Matrices and Covariance Matrices

When relatively little is known about the physical meaning of the dynamics of a control system's state-space model, it is common practice to oversimplify the cost and covariance matrices for LQG controller designs. One method is to assign scalar-weighted identity matrices, as follows:

$$\begin{aligned} R_1 &= I, \quad R_2 = \rho_r I, \quad R_{12} = 0 \\ E\{vv^T\} &= I, \quad E\{nn^T\} = \rho_v I, \quad E\{vn^T\} = 0 \end{aligned} \tag{5.1}$$

where v and n represent the process and sensor noise vectors, respectively. The idea, then, is to reduce the LQG design process to the selection of two scalar parameters — ρ_r to adjust controller authority, and ρ_v to adjust the tracking speed of the Kalman filter. Although it is generally not possible to know the cost matrices which will give the most desirable response or the covariance matrices which most accurately describe the noise, we may take advantage of what knowledge we have about the simply supported plate under study to take a more meaningful design approach.

A large class of damped flexible structures, including those modeled by finite-element methods, can be expressed in *spatial coordinates* by the following differential equation:

$$M\ddot{q}(t) + C\dot{q}(t) + Kq(t) = f(t) \tag{5.2}$$

where q denotes the n -dimensional spatial displacement vector and f denotes the

n -dimensional applied force vector. M , C , and K are symmetric *mass*, *damping*, and *stiffness* matrices, respectively. Assuming the structure has *proportional damping* (i.e., $C = k_1 M + k_2 K$, for some scalar constants k_1 and k_2), the matrices M , C , and K can be simultaneously diagonalized by a left and right multiplication. In other words, the system may be decoupled into *modal coordinates* by means of a variable substitution, so that

$$m_i \ddot{x}_i(t) + c_i \dot{x}_i(t) + k_i x_i(t) = f_i(t), \quad i = 1, 2, \dots, n \quad (5.3)$$

where x_i represents the "displacement" of the i^{th} mode. The proportional damping assumption tends to hold, for all practical purposes, when the damping is very light, as is the case for many flexible structures and in particular for the one studied in this work. Also, the identification procedure used to obtain a model of the simply supported plate assumes a modal system. In terms of modal natural frequencies and damping ratios, ω_i and ζ_i , respectively, an equivalent expression for (5.3) is

$$\ddot{x}_i(t) + 2\zeta_i \omega_i \dot{x}_i(t) + \omega_i^2 x_i(t) = u_i(t), \quad i = 1, 2, \dots, n \quad (5.4)$$

Then the modal frequencies are expressed in terms of the diagonalized mass and stiffness matrices by

$$\omega_i^2 = \frac{k_i}{m_i} \quad (5.5)$$

The energy contained in each mode is the sum of the modal potential and kinetic energies,

$$E_i = \frac{1}{2} k_i x_i^2 + \frac{1}{2} m_i \dot{x}_i^2 \quad (5.6)$$

Therefore,

$$E_i \propto \omega_i^2 x_i^2 + \dot{x}_i^2 \quad (5.7)$$

Rather than penalizing all of the states equally, as in (5.1), it seems reasonable to penalize the total energy in the system. Then the controls may be penalized as before,

or individually, to balance the tradeoff between modal energy and control effort. If we order the displacements and velocities into a state vector, $\bar{x} \triangleq [x^T \dot{x}^T]^T$, this is accomplished by the cost matrices,

$$R_1 = \begin{bmatrix} \Omega^2 & 0 \\ 0 & I \end{bmatrix}, \quad R_2 = \text{diag}\{\rho_{r_1}, \rho_{r_2}, \dots, \rho_{r_m}\}, \quad R_{12} = 0 \quad (5.8)$$

for a system with m controls, where $\Omega \triangleq \text{diag}\{\omega_1, \omega_2, \dots, \omega_n\}$. For the system under study there is only one control input, so this technique leaves only one parameter to adjust. In section 5.2 it will be shown how the cross-weighting matrix, R_{12} , may be manipulated to suit other design goals.

The covariance matrices can also be given a more sensible structure, although the modifications involved may not be significant enough to gain much advantage. For controller design purposes the state and output equations for the plant are expressed in modal coordinates:

$$\begin{aligned} \dot{\bar{x}} &= A\bar{x} + Bu + G_1 w \triangleq A\bar{x} + Bu + [g_{11} \ g_{12}] \begin{bmatrix} v \\ n \end{bmatrix} \\ y &= C\bar{x} + Du + G_2 w \triangleq C\bar{x} + Du + [g_{21} \ g_{22}] \begin{bmatrix} v \\ n \end{bmatrix} \end{aligned} \quad (5.9)$$

It is assumed here that the process and sensor noise are independent of each other. Hence, we have partitioned the exogenous noise vector w into process noise v and sensor noise n . Since sensor noise would only occur in the output equation, $g_{12} = 0$. Also, w is assumed normalized by G_1 and G_2 such that $E\{ww^T\} = I$.

One might arbitrarily assume that the elements of n each affect one modal output in y and to the same degree, so that $g_{22} = \rho_v^{1/2} I$. However, any judgment about the relative intensities and distribution of the sensor noise should be made in *sensor coordinates*. Denote the sensor measurements as a sum of a true signal and sensor noise, as follows:

$$y_{\text{meas}} = y_s + g_s n_s \quad (5.10)$$

The relationship between the sensor outputs and the modal outputs is given by

$$y_{\text{meas}} = \Phi y \quad (5.11)$$

where the *sensor eigenvector* (or *modeshape*) *matrix* Φ has at least as many rows as columns and indicates the relative participation each modal output has on each sensor. The output equation in modal coordinates is approximated by the least squares solution,

$$y = (\Phi^T \Phi)^{-1} \Phi^T y_{\text{meas}} = \Phi^* y_s + \Phi^* g_s n_s \quad (5.12)$$

where Φ^* is the least squares pseudoinverse of Φ , or simply the inverse if Φ is square. Comparison of (5.12) with the output equation in (5.9) reveals that simply setting $g_{22} = \rho_v^{1/2} I$ ignores the structure of the modeshape matrix. Instead, for $E\{n_s n_s^T\} \triangleq I$ we might reasonably set $g_s = \rho_v^{1/2} I$ if all of the sensors are identical and function independently of one another. That gives us

$$g_{22} n = \Phi^* g_s n_s = \rho_v^{1/2} \Phi^* n_s \quad (5.13)$$

The original sensor noise vector n should be replaced by the possibly higher dimensional vector n_s , giving

$$g_{22} = \rho_v^{1/2} \Phi^* \quad (5.14a)$$

but that does not affect the dimension of $V_2 \triangleq G_2 G_2^T = g_{21} g_{21}^T + g_{22} g_{22}^T$. Therefore,

$$V_2 = g_{21} g_{21}^T + \rho_v \Phi^* \Phi^{*T} = g_{21} g_{21}^T + \rho_v (\Phi^T \Phi)^{-1} \quad (5.15)$$

Note that the same result for V_2 may be arrived at by replacing (5.14a) with

$$g_{22} = \rho_v^{1/2} (\Phi^T \Phi)^{-1/2} \quad (5.14b)$$

This alteration has no effect on $V_{12} \triangleq G_1 G_2^T$ either because, as was already indicated, $g_{12} = 0$. Using (5.14b) allows us to retain the lower dimensional sensor noise vector n .

5.2 Disturbance Cancellation and Inverse Optimal Control

When the process noise disturbance is expected to lie in some limited frequency range, it is a highly inefficient allocation of control effort to attempt to reject that disturbance by merely lowering the control penalty, ρ_r , and sensor noise covariance, ρ_v , until the desired degree of rejection is achieved. That procedure results in greater disturbance rejection at all frequencies and potentially requires an unacceptable amount of control effort (as well as excessive sensitivity to sensor noise) to achieve the specified level of rejection in the anticipated disturbance frequency band. Rather than raise controller authority at all frequencies, it is more desirable to concentrate the control effort where it is needed.

Suppose we have the following plant and colored disturbance model, respectively:

$$\begin{aligned}\dot{x}_p &= Ax_p + Bu + G_w w \\ y &= Cx_p + Du + g_{21}w + g_{22}n\end{aligned}\tag{5.16}$$

$$\begin{aligned}\dot{x}_w &= A_w x_w + B_w v \\ w &= C_w x_w\end{aligned}\tag{5.17}$$

Augmenting the plant with the shaping filter dynamics yields

$$\begin{aligned}\frac{d}{dt}\begin{bmatrix} x_p \\ x_w \end{bmatrix} &= \begin{bmatrix} A & G_w C_w \\ 0 & A_w \end{bmatrix} \begin{bmatrix} x_p \\ x_w \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ B_w \end{bmatrix} v \\ y &= \begin{bmatrix} C & g_{21} C_w \end{bmatrix} \begin{bmatrix} x_p \\ x_w \end{bmatrix} + Du + g_{22}n\end{aligned}\tag{5.18}$$

If we assume complete knowledge of the augmented state, the feedback

$$u = -Fx \triangleq -\begin{bmatrix} F_p & F_w \end{bmatrix} \begin{bmatrix} x_p \\ x_w \end{bmatrix}\tag{5.19}$$

produces the closed-loop system and output matrices,

$$A_{cl} = \begin{bmatrix} A - BF_p & G_w C_w - BF_w \\ 0 & A_w \end{bmatrix} \quad (5.20)$$

$$C_{cl} = \begin{bmatrix} C - DF_p & g_{21} C_w - DF_w \end{bmatrix}$$

If g_{21} is nonzero, there are two components of the disturbance — one which enters through G_w and influences the states, and the other which enters by way of g_{21} and feeds through to the outputs. Both components are completely canceled if F_w is chosen such that

$$G_w C_w - BF_w = 0 \quad (5.21a)$$

$$g_{21} C_w - DF_w = 0 \quad (5.21b)$$

Both conditions cannot in general be met simultaneously. However, the type of problem under study has a favorable structure in this regard. For the n_m -mode flexible structure model studied in chapters 6 and 7, the control input and disturbance are forces and affect only the derivatives of the velocity states. Also, the measurements are modal accelerations. Therefore, the matrices B , G_w , D , and g_{21} take the form,

$$B = \begin{bmatrix} 0 \\ b \end{bmatrix}, \quad G_w = \begin{bmatrix} 0 \\ g \end{bmatrix} \quad (5.22)$$

$$D = b, \quad g_{21} = g$$

and both (5.21a) and (5.21b) reduce to:

$$\boxed{bF_w = gC_w} \quad (5.23)$$

A solution, F_w , that satisfies (5.23) cannot be found in general unless the $n_m \times m$ matrix b has rank n_m . This condition requires that the number of actuators be at least as great as the number of modes in the plant model (i.e., $m \geq n_m$). If $m < n_m$ it may not be possible to completely cancel the disturbance, but we can still choose F_w to cancel the effect of the disturbance on m of the modes by selecting the corresponding m rows of b and g to replace the full matrices in (5.23).

If the state feedback (5.19) must pass through a low pass smoothing filter before entering the plant through B , the above technique may still be applied, but it is no longer possible to cancel the disturbance at all frequencies simultaneously. Define the smoothing filter transfer function, $H_l(s) = C_l(sI - A_l)^{-1}B_l$. Then the condition on F_w which produces exact cancellation of the disturbance at frequency ω_d is:

$$bC_l(j\omega_d I - A_l)^{-1}B_l F_w x_w(j\omega_d) = gC_w x_w(j\omega_d) \quad (5.24)$$

The vector $x_w(j\omega_d)$ is found by applying a sinusoid of frequency ω_d to the input of the disturbance shaping filter (5.17) and measuring the response of x_w in the frequency domain. Assuming v is a scalar, as is the case in our single disturbance input model, we set

$$v(t) = \sin(\omega_d t) \quad (5.25)$$

Then, after taking the Fourier transform of (5.17) and solving for x_w we have

$$x(j\omega_d) = (j\omega_d I - A_w)^{-1}B_w v(j\omega_d) \quad (5.26)$$

Substituting (5.26) into (5.24) and eliminating the scalar $v(j\omega_d)$ from both sides of the equation, we arrive at:

$$bC_l(j\omega_d I - A_l)^{-1}B_l F_w (j\omega_d I - A_w)^{-1}B_w = gC_w (j\omega_d I - A_w)^{-1}B_w \quad (5.27)$$

When estimator-based feedback is necessary, infinite disturbance rejection is no longer possible, but the disturbance cancellation technique may still be used to provide good rejection if the estimates of the disturbance states are accurate. Therefore, the effectiveness of this technique depends on the accuracy of our knowledge of the plant (particularly at the disturbance frequency, where the control effort is concentrated) and on the speed of the estimator.

Conceptually, disturbance cancellation depends on the separability of the estimator and regulator designs. The method just discussed assumes that the estimates of the disturbance states are available for feedback so that the corresponding regulator gains can be designed to cancel the disturbance. This restriction presents a problem

when the plant model involves multiplicative white noise (as with the time-domain technique of Chapter 3) or when an optimal reduced order controller (see Chapter 4) is desired. In both cases, the separation principle of LQG does not hold, because the controller solution involves *coupled* Riccati- and Lyapunov-type equations [see (4.16)–(4.19)]. As will be discussed in section 5.5, however, its application is most desirable under precisely these conditions.

Fortunately, disturbance cancellation can be adapted to problems involving a coupled regulator and estimator by applying an idea based on inverse optimal control. The inverse problem of optimal control for linear state feedback problems was first investigated by Kalman (1964). Given a linear state feedback law, $u = -Fx$, the inverse problem is concerned with finding all cost functionals,

$$J = \frac{1}{2} \int_0^{\infty} [x(t)^T R_1 x(t) + 2x(t)^T R_{12} u(t) + u(t)^T R_2 u(t)] dt \quad (5.28)$$

for which the control law is optimal. That is, given F , find all corresponding matrix triples $\{R_1, R_{12}, R_2\}$. Returning to the disturbance cancellation problem, F is considered known, because we can set $F_p = 0$ and solve (5.23) or (5.27) for F_w , assuming for the moment that we have a standard LQG problem. By applying inverse optimal control, we can then solve for cost matrices — R_1 , R_{12} , and R_2 — which provide us with a regulator to cancel the anticipated disturbance. These cost matrices may then be used to design a controller with coupled regulator and estimator for problems involving multiplicative white noise or optimal reduced-order control.

Kalman resolved the inverse optimal control problem for single-input systems, with the restriction: $R_{12} = 0$. He showed (Kalman 1964, Theorem 6) that a solution, $\{R_1, R_2\}$, exists if and only if a certain condition on the return difference function — now known as the *Kalman inequality* — holds. The restriction on R_{12} is not a concern for many problems, because any crossweighting may effectively be eliminated by a suitable modification of the system matrix and the addition of artificial state feedback to compensate for that modification. In estimator-based feedback systems where the separation principle does not hold, however, that method of formulating an equivalent problem is no longer valid.

Kreindler and Jameson (1972) showed that if a nonzero crossweighting is allowed, the inverse problem always has a solution. Since they were concerned with the

conventional problem of finding *all* solutions, $\{R_1, R_{12}, R_2\}$, their approaches are so general (e.g., allowing Riccati solutions not to be positive definite) that they make it difficult to find a single solution.

When only a single solution is sought, as is the case here, the inverse optimal control problem is much simpler. Let $R_2 = I$ and assume that the state feedback law which provides complete disturbance cancellations given by: $u = -Fx$. Then the integrand of the cost functional (5.28) can be rewritten as follows:

$$\begin{aligned} x^T R_1 x + 2x^T R_{12} u + u^T R_2 u & \quad (5.29) \\ &= x^T (R_1 - R_{12} R_2^{-1} R_{12}^T) x + (u + R_2^{-1} R_{12}^T x)^T R_2 (u + R_2^{-1} R_{12}^T x) \\ &= x^T (R_1 - R_{12} R_{12}^T) x + x^T (R_{12}^T - F)^T (R_{12}^T - F) x \end{aligned}$$

Since a negative cost is physically impossible, we must assume that

$$R_1 - R_{12} R_2^{-1} R_{12}^T = R_1 - R_{12} R_{12}^T \geq 0 \quad (5.30)$$

where " ≥ 0 " denotes positive semidefiniteness. Therefore, a set of cost matrices for which F minimizes the cost functional (in fact makes it zero) is:

$$\boxed{\{R_1, R_{12}, R_2\} = \{F^T F, F^T, I\}} \quad (5.31)$$

This means that in order to apply the disturbance cancellation technique to multiplicative white noise or optimal reduced-order control problems, we can use the following algorithm:

- (1) Solve (5.23) or (5.27) for F_w
- (2) Arbitrarily set $F_p = 0$, so that $F = [0 \quad F_w]$
- (3) Compute the desired cost matrices according to (5.31)
- (4) Design the controller using the cost matrices from step (3)

As in the case of LQG problems, the effectiveness of this algorithm depends, in a sense, on a good estimator, since accurate knowledge of the states was assumed. Even though there is no explicit estimator in these modified LQG controllers, the algorithm stated above has worked well consistently for "large" V_1 (corresponding to a fast estimator).

In the event that some transient suppression is desired in addition to good steady disturbance rejection, the algorithm may be modified by adding a penalty to the plant states. The above algorithm yields

$$R_1 = \begin{bmatrix} 0 & 0 \\ 0 & F_w^T F_w \end{bmatrix} \quad (5.32)$$

The desired transient suppression is accomplished by replacing the zero submatrix of the (1,1) block with a sufficiently "large" positive semidefinite matrix. That matrix may be chosen by penalizing modal energy as in section 5.1 (i.e., by choosing some multiple of R_1 from (5.8) to fill the (1,1) block). Experience has shown this modification to the algorithm to be very effective.

5.3 Discrete-Time Controller Design

Continuous-time controllers tend to be more convenient for theoretical work and for frequency-domain analysis, but implementation of a control law on a digital controller requires the design of a discrete-time controller. Ideally, the discrete-time controller is directly designed from sampled-data model of the control system. Since discrete-time design algorithms are readily available for the standard LQG problem, this direct design approach was used to implement controllers on the hardware (see Chapter 7) when neither multiplicative white noise nor optimal projection was involved. The development of the sampled-data version of those modified LQG problems, as well as the algorithms necessary to solve for their respective compensators, is beyond the scope of this work and arguably could be considered a duplication of effort. A good approximation to the directly designed discrete-time controller can be obtained by discretizing the corresponding continuous-time controller. Of course, the approximation is particularly good when the sample rate is high. Therefore, it is fortunate that optimal reduced-order controllers tend to make higher sample rates possible.

The remainder of this section discusses two considerations which should be taken into account in designing a discrete-time controller.

A. Conversion of Cost and Covariance Matrices to Discrete-Time

In sampled-data controller design, the continuous-time model of an actual

continuous-time plant is translated into a discrete-time equivalent, and there is relatively little intuitive understanding for the meaning of the resulting discrete-time variables. Therefore, rather than designing the compensator entirely in discrete-time, it makes more sense to specify the cost functional and noise covariances in continuous-time and then translated them to discrete-time for computing the optimal discrete-time controller. Assume the continuous-time plant model,

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) + w_1(t) \\ y(t) &= Cx(t) + Du(t) + w_2(t)\end{aligned}\tag{5.33}$$

where the noise covariances are given by:

$$\mathbb{E}\left\{\begin{bmatrix} w_1(t) \\ w_2(t) \end{bmatrix} \begin{bmatrix} w_1^T(t) & w_2^T(t) \end{bmatrix}\right\} = \begin{bmatrix} V_1 & V_{12} \\ V_{12}^T & V_2 \end{bmatrix} \delta(t) \triangleq V \delta(t)\tag{5.34}$$

And define the zero-order hold equivalent of the plant:

$$\begin{aligned}x_{k+1} &= \Phi(\Delta t)x_k + \Gamma(\Delta t)u_k + w_{1k} \\ y_k &= Cx_k + Du_k + w_{2k}\end{aligned}\tag{5.35}$$

where Δt is the sampling interval, the subscript " k " denotes the sampling instant for $k = 0, 1, 2, \dots$ (i.e., $x_k \triangleq x(t_k)$, where the k^{th} sampling instant occurs at time t_k), and

$$\Phi(\Delta t) = e^{A \cdot \Delta t}\tag{5.36}$$

$$\Gamma(\Delta t) = \int_0^{\Delta t} \Phi(t) dt \cdot B$$

$$w_{1k} = \int_{t_k}^{t_{k+1}} \Phi(t - t_k) w_1(t) dt\tag{5.37}$$

$$w_{2k} = w_2(t_k)$$

Then the discrete-time equivalent to the continuous-time cost functional, (5.28), takes the form,

$$J = \frac{1}{2} \sum_{k=0}^{\infty} [x_k^T \hat{R}_1 x_k + 2x_k^T \hat{R}_{12} u_k + u_k^T \hat{R}_2 u_k]\tag{5.38}$$

where the discrete-time cost matrices are denoted with carats and are given by (Stengel 1986, pp. 276-7):

$$\begin{aligned}\hat{R}_1 &= \int_0^{\Delta t} \Phi^T(t) R_1 \Phi(t) dt \\ \hat{R}_{12} &= \int_0^{\Delta t} \Phi^T(t) [R_1 \Gamma(t) + R_{12}] dt \\ \hat{R}_2 &= \int_0^{\Delta t} [\Gamma^T(t) R_1 \Gamma(t) + \Gamma^T(t) R_{12} + R_{12}^T \Gamma(t) + R_2] dt\end{aligned}\tag{5.39}$$

The discrete-time equivalent covariance matrices are computed by their definitions, using (5.34) and (5.37) as follows:

$$\begin{aligned}\hat{V}_1 &\triangleq \mathbb{E} [w_{1k} w_{1k}^T] \\ &= \mathbb{E} \left[\int_{t_k}^{t_k + \Delta t} \Phi(t - t_k) w_1(t) dt \cdot \int_{t_k}^{t_k + \Delta t} w_1^T(t) \Phi^T(t - t_k) dt \right] \\ &= \int_{t_k}^{t_k + \Delta t} \int_{t_k}^{t_k + \Delta t} \Phi(t - t_k) \mathbb{E} [w_1(t) w_1^T(t)] \Phi^T(\tau - t_k) d\tau dt \\ &= \int_{t_k}^{t_k + \Delta t} \Phi(t - t_k) \int_{t_k}^{t_k + \Delta t} V_1 \delta(t - \tau) \Phi^T(\tau - t_k) d\tau dt \\ &= \int_{t_k}^{t_k + \Delta t} \Phi(t - t_k) V_1 \Phi^T(t - t_k) dt \\ &= \int_0^{\Delta t} \Phi(t) V_1 \Phi^T(t) dt\end{aligned}\tag{5.40}$$

$$\begin{aligned}\hat{V}_{12} &\triangleq \mathbb{E} [w_{1k} w_{2k}^T] \\ &= \mathbb{E} \left[\int_{t_k}^{t_k + \Delta t} \Phi(t - t_k) w_1(t) dt \cdot w_2^T(t_k) \right] \\ &= \int_{t_k}^{t_k + \Delta t} \Phi(t - t_k) \mathbb{E} [w_1(t) w_2^T(t_k)] dt \\ &= \int_{t_k}^{t_k + \Delta t} \Phi(t - t_k) V_{12} \delta(t - t_k) dt \\ &= V_{12}\end{aligned}\tag{5.41}$$

$$\begin{aligned}
\hat{V}_2 &\triangleq \mathbf{E} [w_{2k} w_{2k}^T] \\
&= \mathbf{E} [w_2(t_k) w_2^T(t_k)] \\
&= V_2 \delta(0) \quad ???
\end{aligned} \tag{5.42}$$

From (5.42) we see that the literal translation to discrete-time results in an infinite sensor noise covariance. If (5.42) were used for the controller design computations, the Kalman filter would degenerate to an open circuit. The problem arises from the fact that the δ -correlated, infinite-power white noise model assumed by LQG theory does not reflect reality. Intuitively, it seems desirable to retain the structure of the covariance derived in (5.42), while changing the scalar multiplier to a finite number. Gelb (1984, p.121) argues in favor of using

$$\boxed{\hat{V}_2 = \frac{V_2}{\Delta t}} \tag{5.43}$$

because the resulting discrete-time sensor noise covariance approaches the continuous-time covariance as $\Delta t \rightarrow 0$. That was the method used for this study, although the actual scalar multiplier used is not important, since it tends to become a design parameter in practice.

The matrix integrals used to compute the discrete-time cost and covariance matrices are easily evaluated numerically — for example, by means of the forward rectangular rule. Only 20 to 30 intervals were required for good accuracy in the examples of this study, but fewer are necessary for smaller Δt .

B. Computational Delay

Standard LQG compensators (i.e., those with a Kalman filter separate from the regulator) are easily modified to overcome the computational delay inherent in discrete-time controllers. For a one-sampling-interval time delay, the Kalman filter extrapolation and update equations need merely to be reordered to turn the Kalman *filter* into a one-step-ahead *predictor*. The predicted state estimates are then fed back, by means of the regulator, so that the proper state estimates reach the actuators at the proper time.

In problems involving multiplicative white noise and/or optimal projection, the lack of an explicit estimator makes that procedure impossible. If the continuous-time controller is simply discretized, there will be a one-interval time lag between the time of the intended controller command and the instant that command reaches the actuators. For faster sample rates, the problem is smaller and may not be significant. In case the computational delay is significant, it should be modeled in the control system.

A pure one-interval delay is precisely modeled by inserting the irrational transfer function

$$H_D(s) = e^{-s \cdot \Delta t} \quad (5.44)$$

between the commanded control signal leaving the controller and the actual control input entering the plant. However, since $H_D(s)$ is infinite-dimensional, a suitable finite-dimensional approximation must be found — preferably of as low order as possible, because additional dynamics tend to complicate controller design computations and increase controller order. Padé approximations to $e^{-s \cdot \Delta t}$ are useful in this context. A Padé approximation tends to be more accurate than the Taylor series approximation of the same order, and it is easily computed (Bender and Orszag 1978, secs. 8.3 and 8.4). As an example, the second-order Padé approximation to (5.44) is given here:

$$\begin{aligned} \hat{H}_D(s) &= \frac{1 - \frac{1}{2}s \cdot \Delta t + \frac{1}{12}(s \cdot \Delta t)^2}{1 + \frac{1}{2}s \cdot \Delta t + \frac{1}{12}(s \cdot \Delta t)^2} \\ &= \frac{s^2 - \frac{6}{\Delta t}s + \frac{12}{\Delta t^2}}{s^2 + \frac{6}{\Delta t}s + \frac{12}{\Delta t^2}} \end{aligned} \quad (5.45)$$

5.4 Iterative Relaxation Algorithm for Solving Coupled Riccati/Lyapunov Equations

As mentioned in section 4.1, problems involving multiplicative white noise or optimal controller order reduction require an iterative method to find a solution to the coupled extremal equations. A number of different algorithms are described in the literature, but they all belong to one of two categories (or are a hybrid of the two) — *iterative relaxation* methods and *continuation* (or *homotopy*) methods. The idea behind the iterative relaxation approach is to alternately solve the coupled equations for one unknown at a time while treating the other unknowns as constants. The main

advantage of this method of solution is that it requires no prior analytical development, assuming standard Riccati- and Lyapunov-equation solvers are already available, and it is therefore easily and quickly implemented. The main disadvantage is that this type of algorithm is not guaranteed to converge to a solution.

The iterative relaxation algorithm detailed in this section is a variation on one described by Bernstein and Hyland (1988b, pp. 290-91). The algorithm contains two nested loops and is designed to handle the general reduced-order controller problem in the presence of simultaneous state-, control-, and measurement-dependent noise. More specifically, it attempts to solve the general optimal projection equations, (4.16)–(4.20), for Q , P , \hat{Q} , \hat{P} , and τ_{\perp} and subsequently to find the controller state-space matrices — A_c , B_c , and C_c — using (4.15). If a full-order controller is desired, the algorithm is easily modified by removing the outer loop.

Define the *absolute norm* of an $m \times n$ matrix M as follows:

$$\|M\|_A \triangleq \max(M_{ij}, i = 1, 2, \dots, m; j = 1, 2, \dots, n) \quad (5.46)$$

The iterative relaxation algorithm used in this study will now be stated:

- (1) If the Stratonovich multiplicative noise interpretation is desired, modify A , B , and C using the substitutions prescribed in (3.73).
- (2) Perform initializations:
 - (a) Let $\gamma_i = 0$ ($i = 1, 2, \dots, p$) and $\tau = I_n$ (i.e., $\tau_{\perp} = 0$).
 - (b) Solve Riccati equations (4.16) and (4.17) for initial values of Q and P .
 - (c) Solve Lyapunov equations (4.18) and (4.19) for initial \hat{Q} and \hat{P} .
 (Note: this initialization corresponds to the standard LQG solution.)

Beginning of outer loop:

(3) Update the optimal projection matrix, τ :

(a) Compute a balancing transformation, Ψ , such that:

$$\Psi^{-1}\hat{Q}\Psi^{-1} = \Psi^T\hat{P}\Psi = \Lambda \quad (5.47)$$

$$\Lambda \triangleq \text{diag}\{\lambda_1, \lambda_2, \dots, \lambda_n\}$$

(Laub 1980). Note: all λ_i ($i = 1, 2, \dots, n$) are nonnegative real.

(b) Perform a basis rearrangement:

Rearrange the λ_i so that $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$, then rearrange the corresponding n columns of Ψ using that same ordering.

(c) Compute

$$\tau = \Psi \begin{bmatrix} I_{n_c} & 0 \\ 0 & \gamma \end{bmatrix} \Psi^{-1} \quad (5.48)$$

where

$$\gamma \triangleq \text{diag}\left\{\frac{\lambda_{n_c+1}}{\lambda_{n_c}}, \frac{\lambda_{n_c+2}}{\lambda_{n_c}}, \dots, \frac{\lambda_n}{\lambda_{n_c}}\right\} \quad (5.49)$$

(4) Test for convergence of Q , P , \hat{Q} , \hat{P} , and τ (condition to leave outer loop):

(a) Compute relative errors, as follows.

$$e_Q \triangleq \|\text{r.h.s. of (4.16)}\|_A / \|V_1\|_A \quad (5.50)$$

$$e_P \triangleq \|\text{r.h.s. of (4.17)}\|_A / \|R_1\|_A$$

$$e_{\hat{Q}} \triangleq \|\text{r.h.s. of (4.18)}\|_A / \|\mathcal{Q}V_2^{-1}\mathcal{Q}^T\|_A$$

$$e_{\hat{P}} \triangleq \|\text{r.h.s. of (4.19)}\|_A / \|\mathcal{P}^T R_2^{-1} \mathcal{P}\|_A$$

(b) If $\max\{e_Q, e_P, e_{\hat{Q}}, e_{\hat{P}}\} \leq \epsilon_1$ (the outer loop tolerance), go to (14).

Otherwise, go to (5).

Beginning of inner loop:

(5) Update \hat{V}_2 , \hat{R}_2 , \mathcal{Q} , and \mathcal{P} using equations (3.53), (3.58), (3.63), and (3.64).

(6) Update Q :

Solve Riccati equation (4.16) for Q by treating the terms involving γ_i and τ_{\perp} as constants, effectively adding them to V_1 . The γ_i term in the definition of \mathcal{Q} should be added to V_{12} , effectively defining a modified cross-covariance.

(7) Repeat (5).

(8) Update P :

Solve Riccati equation (4.17) for P by treating the terms involving γ_i and τ_{\perp} as constants, effectively adding them to R_1 , with one exception. The γ_i term in the definition of \mathcal{P} should be added to R_{12} , effectively defining a modified cross-weighting.

(9) Repeat (5) and update A_P using equation (3.65).

(10) Update \hat{Q} :

Solve Lyapunov equation (4.18) for \hat{Q} by treating the term involving τ_{\perp} as a constant, effectively adding it to the other constant term.

(11) Repeat (5) and update A_Q using equation (3.65).

(12) Update \hat{P} :

Solve Lyapunov equation (4.19) for \hat{P} by treating the term involving τ_{\perp} as a constant, effectively adding it to the other constant term.

(13) Test for convergence of Q , P , \hat{Q} , and \hat{P} (condition to leave inner loop):

(a) Compute maximum relative change since the last iteration, as follows.

$$\Delta Q \triangleq \|Q^i - Q^{i-1}\|_A / \|Q^i\|_A \quad (5.51)$$

$$\Delta P \triangleq \|P^i - P^{i-1}\|_A / \|P^i\|_A$$

$$\Delta \hat{Q} \triangleq \|\hat{Q}^i - \hat{Q}^{i-1}\|_A / \|\hat{Q}^i\|_A$$

$$\Delta \hat{P} \triangleq \|\hat{P}^i - \hat{P}^{i-1}\|_A / \|\hat{P}^i\|_A$$

where the superscripts indicate the iteration number of the inner loop — i being the current iteration.

- (b) If $\max\{Q, P, \hat{Q}, \hat{P}\} \leq \epsilon_2$ (the inner loop tolerance), go to (3).
Otherwise, go to (5).

(14) Compute G and Γ using equations (4.14).

(15) Compute A_c , B_c , and C_c using equations (4.15).

We should recall that the optimal projection equations are *necessary* conditions for the solution of optimal reduced-order controllers. According to Hyland and Bernstein (1988b, p. 292), for an n^{th} -order plant with m control inputs and l sensor outputs, there is only one solution if $n_c \geq \min(n, m, l)$, but otherwise there may be as many as

$$\binom{\min(n, m, l)}{n_c} \triangleq \frac{[\min(n, m, l)]!}{[\min(n, m, l) - n_c]! \cdot n_c!}$$

solutions. The choice of initial values for Q , P , \hat{Q} , and \hat{P} corresponding to the LQG solution in step (2) is an attempt to begin the iterations as close as possible to the optimal solution sought, therefore minimizing the risk of convergence to a suboptimal solution and reducing computation time. However, for the problems treated in this study, $m = 1$. That allowed the controller order to be specified as low as needed without concern for the existence of multiple solutions to the optimal projection equations.

The most difficult parts of the algorithm to establish are the convergence tests and tolerance specifications. These are important, since a test which is too stringent will never be satisfied, and a test which is too relaxed will not allow the algorithm to converge. In either case, the algorithm becomes an infinite loop. Unfortunately, the tolerances must be adjusted to suit the problem at hand, because the values of the most effective tolerances are somewhat sensitive to the plant model and even to controller authority. Although both tests are in some sense normalized, they are not universally applicable with the same tolerances. When the algorithm is repeatedly applied to the same plant model, acceptable tolerances can be chosen after some experimentation. Otherwise, it is advisable to do one or more of the following: (1) monitor more than one indicator, (2) make the tolerances automatically adaptive, or (3) give the software the

capability to accept manual changes in the tolerances between iterations.

The convergence test used for the inner loop — step (12) — was chosen because it is very cheap computationally to monitor *changes* in values since the last iteration. This type of test has also proven to be quite effective in determining convergence, even when compared to more sophisticated tests. The outer loop convergence test — step (4) — was suggested by Richter and Collins (1989). It seemed to be the logical choice for the outer loop (or for the *only* loop if both are not present), because it monitors the equation errors directly, although it requires much more matrix arithmetic. Still another convergence test for the outer loop was suggested by Hyland and Bernstein (1988b, p. 291):

$$\frac{\text{trace}(\tau) - n_c}{n_c} < \epsilon \quad (5.52)$$

for some tolerance ϵ . This test is based on the property — $\text{trace}(\tau) = n_c$ — when complete convergence has been attained. To see why this property holds, recall equations (4.13) and (4.14), and note that

$$\tau = G^T \Gamma, \quad \Gamma G^T = I_{n_c} \quad (5.53)$$

Then

$$\text{trace}(\tau) = \text{trace}(G^T \Gamma) = \text{trace}(\Gamma G^T) = n_c \quad (5.54)$$

Step (5) — the update of matrix values which appear in the Riccati- and Lyapunov-type equations — is repeated in several places throughout the algorithm in order to give those equations the most recent information. It may be omitted in some places in order to save a little computation time, but the algorithm may diverge if it is not repeated frequently enough.

5.5 Homotopy Algorithm for Solving Optimal Projection Equations

Several homotopy algorithms have been developed to solve the optimal projection equations, but as of this writing they have only been hinted at (e.g., Richter 1987; Richter and Collins 1989) and have not been published in explicit form in archival journals. For that reason, a homotopy algorithm is developed here, although with the

restriction that it applies to optimal projection only (i.e., multiplicative white noise is not included). A similar algorithm for the more general case may be derived based on the principles given in this section, although it would likely be much more complicated. The presence of multiplicative white noise adds several new terms to the Riccati and Lyapunov equations, resulting in the appearance of all five unknowns (Q , P , \hat{Q} , \hat{P} , and τ) in all four equations when state-, control-, and measurement-dependent noise are present.

Examination of the optimal projection equations (4.7)–(4.11) reveals that the degree of coupling among the Riccati- and Lyapunov-type equations is directly related to the “size” of the matrices V_2 and R_2 . When the sensor noise covariances and control penalties are large, the coupling is reduced. However, when large controller authority (i.e., small V_2 and R_2) is desired, the interaction among these equations tends to be great, and the iterative algorithm of section 5.4 is less likely to converge to a solution. This problem frequently occurred when the disturbance cancellation method of section 5.2 was not used and the resulting controller authority needed to meet disturbance rejection specifications was necessarily high. Rather than alternately solving for Q , P , \hat{Q} , and \hat{P} , repeatedly, the homotopy algorithm replaces this inner loop iteration with numerical integration and finds these solutions (for fixed τ) in a single pass. It does so by continuously deforming the Riccati equation solutions from a known solution (e.g., the LQG solution) into the solution sought.

Homotopy methods of solving complex equations, such as the one described here, are developed in three steps: (1) Find the solution to a simple, but related equation; (2) Express the relationship between the solutions to the simple and complex equations in terms of a differential equation; and (3) (Numerically) integrate the differential equation to obtain the solution to the complex equation. Given a function $F: \mathbb{R}^n \rightarrow \mathbb{R}^n$, suppose we seek the solution, u , to $F(u) = 0$. Define a function $H: \mathbb{R}^n \times [0, 1] \rightarrow \mathbb{R}^n$ such that:

$$\begin{aligned} \text{(a)} \quad & H(u(\alpha), \alpha) = 0 \text{ for } \alpha \in [0, 1] \\ \text{(b)} \quad & H(u, 1) = F(u) \\ \text{(c)} \quad & \text{a solution } u(0) \text{ to } H(u(0), 0) = 0 \text{ is known} \end{aligned} \tag{5.55}$$

Then if H is continuous and $(\partial H / \partial u)^{-1}$ exists over the entire interval $\alpha \in [0, 1]$, the solution $u(1)$ to $H(u(1), 1) = F(u) = 0$ may be found as follows. First, differentiate

$H(u(\alpha), \alpha) = 0$ with respect to α using the chain rule for vectors (Graham 1981, sec. 4.3):

$$\frac{dH(u(\alpha), \alpha)}{d\alpha} = \frac{du(\alpha)}{d\alpha} \cdot \frac{\partial H(u(\alpha), \alpha)}{\partial u(\alpha)} + \frac{\partial H(u(\alpha), \alpha)}{\partial \alpha} = 0 \quad (5.56)$$

Then

$$\frac{du(\alpha)}{d\alpha} = -\frac{\partial H(u(\alpha), \alpha)}{\partial \alpha} \left(\frac{\partial H(u(\alpha), \alpha)}{\partial u(\alpha)} \right)^{-1} \quad (5.57)$$

Since $u(0)$ is known, we can treat (5.57) as an initial value problem, integrating $du(\alpha)/d\alpha$ over the interval from 0 to 1 to obtain the solution $u \triangleq u(1)$. For a more complete discussion of homotopy methods, see Richter and DeCarlo (1983).

Before applying the homotopy principle to the modified Riccati equations (4.7)–(4.8), it is advisable to rewrite these equations in vector form in order to avoid the problem of taking derivatives of a matrix with respect to a matrix. Recall equation (4.7):

$$0 = AQ + QA^T - \mathcal{Q}V_2^{-1}\mathcal{Q}^T + V_1 + \tau_{\perp} \mathcal{Q}V_2^{-1}\mathcal{Q}^T \tau_{\perp}^T \quad (4.7)$$

where $\mathcal{Q} \triangleq QC^T + V_{12}$, and assume τ_{\perp} is fixed. Then the equivalent equation in Q that we wish to solve is:

$$\begin{aligned} F_Q(\text{vec} Q, \text{vec} \mathcal{Q}^T) &= (A \oplus A) \text{vec} Q + \text{vec} V_1 - \text{vec}(\mathcal{Q}V_2^{-1}\mathcal{Q}^T) + \tau^* \text{vec}(\mathcal{Q}V_2^{-1}\mathcal{Q}^T) \\ &= 0 \end{aligned} \quad (5.58)$$

where

$$\tau^* \triangleq \tau_{\perp} \otimes \tau_{\perp} \quad (5.59)$$

If the last term in (5.58) [or (4.7)] were not present, we would have a standard Riccati equation, whose solution is easily found. Therefore, a logical choice for the function H is:

$$H_Q(\text{vec}Q, \text{vec}Q^T, \alpha) = (A \oplus A) \text{vec}Q + \text{vec} V_1 - \text{vec}(QV_2^{-1}Q^T) + \alpha\tau^* \text{vec}(QV_2^{-1}Q^T) = 0 \quad (5.60)$$

For $\alpha = 0$, (5.60) is equivalent to the standard Riccati equation, and for $\alpha = 1$, we recover the modified Riccati equation (4.7). Denote the initial solution for Q (i.e., the LQG solution) by Q^0 , corresponding to the constant projection matrix: $\tau^0 = I \Rightarrow \tau^{*0} = 0$. Equation (5.60) is suitable if we wish to solve (4.7) only once, for constant $\tau = \tau^1$, where τ^1 has been computed based on the initial values of \hat{Q} and \hat{P} . However, we also wish to solve for τ , so the outer loop of the algorithm will be updating the projection matrix, giving τ^i , $i = 1, 2, \dots$. Therefore, on the i^{th} iteration we need to continuously deform a known solution Q^{i-1} into Q^i based on our knowledge of τ^i and τ^{i-1} . To that end, consider the function,

$$\begin{aligned} H^i_Q(\text{vec}Q^i, \text{vec}Q^{iT}, \alpha) &= (A \oplus A) \text{vec}Q^i + \text{vec} V_1 - \text{vec}(Q^i V_2^{-1} Q^{iT}) \\ &\quad + \tau^{*i-1} \text{vec}(Q^{i-1} V_2^{-1} Q^{i-1T}) \\ &\quad + \alpha[\tau^{*i} \text{vec}(Q^i V_2^{-1} Q^{iT}) \\ &\quad - \tau^{*i-1} \text{vec}(Q^{i-1} V_2^{-1} Q^{i-1T})] = 0 \end{aligned} \quad (5.61)$$

and note that it satisfies the conditions in (5.55). For $\alpha = 0$, the solution $(\text{vec}Q^{i-1}, \text{vec}Q^{i-1T})$ from the previous iteration satisfies (5.61), and for $\alpha = 1$, (5.61) is equivalent to (5.58). Now we may develop the initial condition problem to solve by taking the derivative of this function with respect to α , as follows:

$$\frac{dH^i_Q}{d\alpha} = \frac{d\text{vec}Q^i}{d\alpha} \cdot \frac{\partial H^i_Q}{\partial \text{vec}Q^i} + \frac{d\text{vec}Q^{iT}}{d\alpha} \cdot \frac{\partial H^i_Q}{\partial \text{vec}Q^{iT}} + \frac{\partial H^i_Q}{\partial \alpha} = 0 \quad (5.62)$$

By the definition of Q , we have:

$$\frac{d\text{vec}Q^{iT}}{d\alpha} = \frac{d}{d\alpha} \text{vec}(CQ^i + V_{12}^T) = \frac{d}{d\alpha} (I \otimes C) \text{vec}Q^i = \frac{d\text{vec}Q^i}{d\alpha} \cdot (I \otimes C^T) \quad (5.63)$$

Therefore,

$$\frac{d\text{vec}Q^i}{d\alpha} \left[\frac{\partial H^i_Q}{\partial \text{vec}Q^i} + (I \otimes C^T) \cdot \frac{\partial H^i_Q}{\partial \text{vec}Q^{iT}} \right] + \frac{\partial H^i_Q}{\partial \alpha} = 0 \quad (5.64)$$

or:

$$\frac{d\text{vec}Q^i}{d\alpha} = -\frac{\partial H_Q^i}{\partial \alpha} \left[\frac{\partial H_Q^i}{\partial \text{vec}Q^i} + (I \otimes C^T) \cdot \frac{\partial H_Q^i}{\partial \text{vec}Q^{iT}} \right]^{-1} \quad (5.65)$$

where

$$\frac{\partial H_Q^i}{\partial \alpha} = \left[\tau^{*i} \text{vec}(Q^i V_2^{-1} Q^{iT}) - \tau^{*i-1} \text{vec}(Q^{i-1} V_2^{-1} Q^{i-1T}) \right]^T$$

$$\frac{\partial H_Q^i}{\partial \text{vec}Q^i} = (A \oplus A)^T$$

$$\frac{\partial H_Q^i}{\partial \text{vec}Q^{iT}} = - \left[(V_2^{-1} Q^{iT} \otimes I)_{(n)} + (I \otimes V_2^{-1} Q^{iT}) \right] (I - \alpha \tau^{*i})^T$$

The properties used to take these partial derivatives may all be found in Graham (1981). The operator denoted by the subscript “(n)” is also borrowed from Graham (1981, p. 71). It has the effect of reordering the rows of a matrix by taking the first row followed by each subsequent n^{th} row, then the second row followed by each subsequent n^{th} row, etc. Here, n is the order of the (augmented) plant (i.e., $A \in \mathbb{R}^{n \times n}$).

In summary, we may solve the Riccati equation (4.7) for Q^i (given a fixed τ^i) by treating the previous solution, Q^{i-1} , as the initial condition at $\alpha = 0$, then integrating (5.65) over the interval from $\alpha = 0$ to $\alpha = 1$. The starting value, Q^0 , is the LQG solution for Q (i.e., the solution for $\tau_{\perp} = 0$).

The same procedure may be applied to the other Riccati equation, (4.8). From the definition of \mathcal{P} (4.6) we have:

$$\frac{d\text{vec}\mathcal{P}^i}{d\alpha} = \frac{d}{d\alpha} \text{vec}(B^T P^i + R_{12}^T) = \frac{d}{d\alpha} (I \otimes B^T) \text{vec}P^i = \frac{d\text{vec}P^i}{d\alpha} \cdot (I \otimes B) \quad (5.66)$$

which leads to:

$$\frac{d\text{vec}P^i}{d\alpha} = -\frac{\partial H_P^i}{\partial \alpha} \left[\frac{\partial H_P^i}{\partial \text{vec}P^i} + (I \otimes B) \cdot \frac{\partial H_P^i}{\partial \text{vec}\mathcal{P}^i} \right]^{-1} \quad (5.67)$$

where

$$\frac{\partial H_P^i}{\partial \alpha} = \left[\tau^{*i\top} \text{vec}(\mathcal{P}^{i\top} R_2^{-1} \mathcal{P}^i) - \tau^{*i-1\top} \text{vec}(\mathcal{P}^{i-1\top} R_2^{-1} \mathcal{P}^{i-1}) \right]^\top$$

$$\frac{\partial H_P^i}{\partial \text{vec}P^i} = A \oplus A$$

$$\frac{\partial H_P^i}{\partial \text{vec}\mathcal{P}^i} = - \left[(R_2^{-1} \mathcal{P}^i \otimes I)_{(n)} + (I \otimes R_2^{-1} \mathcal{P}^i) \right] (I - \alpha \tau^{*i})$$

The Lyapunov equations, (4.9)–(4.10), do not require a homotopy solution for fixed τ^i , because: (a) the variables Q and P can be treated as fixed constants, since they have already been solved for on the first pass, and (b) the term involving the projection matrix is not a function of the only remaining variable, \hat{Q} (or \hat{P}), and may therefore be treated as part of the constant term. That is, having already solved the modified Riccati equations for Q and P , the modified Lyapunov equations may be treated as standard Lyapunov equations and be solved immediately for \hat{Q} and \hat{P} .

The entire homotopy algorithm for the solution of the optimal reduced-order control problem (without multiplicative white noise) is now stated.

(1) Perform initializations:

- (a) Let $\tau^0 = I_n$ (i.e., $\tau_{\perp}^0 = 0$ and $\tau^{*0} = 0$).
 - (b) Solve Riccati equations (4.7) and (4.8) for the initial values, Q^0 and P^0 .
 - (c) Solve Lyapunov equations (4.9) and (4.10) for initial values \hat{Q}^0 and \hat{P}^0 .
 - (d) Set the iteration number: $i = 0$.
- (Note: this initialization corresponds to the standard LQG solution.)

Beginning of main loop:

- (2) Increment the iteration number: $i \leftarrow i + 1$.
- (3) Compute the optimal projection matrix, τ^i (based on \hat{Q}^{i-1} and \hat{P}^{i-1}), exactly as in step (3) of the algorithm in section 5.4.
- (4) Test for convergence of Q^{i-1} , P^{i-1} , \hat{Q}^{i-1} , \hat{P}^{i-1} , and τ^i (condition to leave main loop):
 - (a) Compute relative errors, as follows.

$$e_Q \triangleq \|\text{r.h.s. of (4.16)}\|_A / \|V_1\|_A \quad (5.50)$$

$$e_P \triangleq \|\text{r.h.s. of (4.17)}\|_A / \|R_1\|_A$$

$$e_{\hat{Q}} \triangleq \|\text{r.h.s. of (4.18)}\|_A / \|\Omega V_2^{-1} \Omega^T\|_A$$

$$e_{\hat{P}} \triangleq \|\text{r.h.s. of (4.19)}\|_A / \|\mathcal{P}^T R_2^{-1} \mathcal{P}\|_A$$

- (b) If $\max\{e_Q, e_P, e_{\hat{Q}}, e_{\hat{P}}\} \leq \epsilon$ (the tolerance), go to (12).
Otherwise, go to (5).
- (5) Compute Ω^{i-1} and \mathcal{P}^{i-1} (based on Q^{i-1} and P^{i-1}) using equations (3.63) and (3.64), and compute τ^{*i} using (5.59).
- (6) Compute Q^i :
Numerically integrate (5.65) over the interval from $\alpha = 0$ to $\alpha = 1$.
- (7) Compute P^i :
Numerically integrate (5.67) over the interval from $\alpha = 0$ to $\alpha = 1$.
- (8) Compute Ω^i and \mathcal{P}^i (based on Q^i and P^i) using equations (3.63) and (3.64), and update A_P and A_Q using equations (3.65).
- (9) Compute \hat{Q} :
Solve Lyapunov equation (4.9) for \hat{Q} by treating the term involving τ_{\perp} as a constant, effectively adding it to the other constant term.

(10) Compute \hat{P} :

Solve Lyapunov equation (4.10) for \hat{P} by treating the term involving τ_{\perp} as a constant, effectively adding it to the other constant term.

(11) Go to (4).

(12) Compute G and Γ using equations (4.14).

(13) Compute A_c , B_c , and C_c using equations (4.15).

6. Evaluation of Designs

The controller design principles developed in Chapters 2-5 were applied to an FDLTI model of a simply supported plate in order to study their effectiveness. This chapter describes the complete control system model, then evaluates the comparative value of the various parameter-robust and optimal reduced-order controller designs based on this model. Chapter 7 applies the analytical techniques of this chapter to solve the problems of plant uncertainty and high controller order that were encountered in the actual hardware.

6.1 Problem Description

The simply supported plate is made of cold-rolled steel. It is rectangular, of dimensions $.5\text{m} \times .6\text{m} \times 2.9\text{mm}$, and has attached to it twelve accelerometer sensors on one side and two point force actuators on the other side — one for the control input and one for external disturbance generation. The accelerometers are lightweight piezoelectric devices mounted to the plate with wax and are placed in a 3×4 rectangular array with locations devised to assure observability of the first twelve vibrational modes of the plate. The presence of more sensors than observed modes (i.e., *spatial oversampling*) is a redundancy that tends to provide greater accuracy in the sensor measurements. The actuators are electromagnetic shakers with a magnitude frequency response from voltage input command to force output that is essentially constant over the frequency range of interest. They are placed in locations coinciding with two of the sensor positions and chosen such that most of the authority of these shakers is on the first two modes. Also, the shakers are located near node lines of modes three and four, with the intention of limiting the “spillover” of the control signal into the higher frequency modes. For the experiments carried out in Chapter 7, the disturbance signal was produced by a function generator. The zero-order hold control signal was sent through a second-order low pass filter before reaching the control shaker for the purpose of smoothing the signal and preventing aliasing. For more details concerning the hardware configuration of the simply supported plate experiment, see Rubenstein (1991).

The plate with actuators and sensors is described by the standard state-space model,

$$\begin{aligned}\dot{x}_p(t) &= A_p x_p(t) + B_p u(t) + [g_{11} \ g_{12}] w_p(t) \\ y(t) &= C_p x_p(t) + D_p u(t) + [g_{21} \ g_{22}] w_p(t)\end{aligned}\quad (6.1)$$

Expanding the matrices in (6.1), we define the modal model,

$$\begin{aligned}\frac{d}{dt} \begin{bmatrix} x_p(t) \\ x_v(t) \end{bmatrix} &= \begin{bmatrix} 0 & I \\ -\Omega^2 & -2Z\Omega \end{bmatrix} \begin{bmatrix} x_p(t) \\ x_v(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \Phi_u \end{bmatrix} u(t) + \begin{bmatrix} 0 & 0 \\ \Phi_w & 0 \end{bmatrix} \begin{bmatrix} w_1(t) \\ w_2(t) \end{bmatrix} \\ y(t) &= \begin{bmatrix} -\Omega^2 & -2Z\Omega \end{bmatrix} \begin{bmatrix} x_p(t) \\ x_v(t) \end{bmatrix} + \Phi_u u + \begin{bmatrix} \Phi_w & \rho_v^{1/2} (\Phi^T \Phi)^{-1/2} \end{bmatrix} \begin{bmatrix} w_1(t) \\ w_2(t) \end{bmatrix}\end{aligned}\quad (6.2)$$

The time variables and their dimensions are defined as follows:

$$\begin{aligned}x_p &\in \mathbb{R}^{n_m} \dots \dots \dots \text{Modal position states} \\ x_v &\in \mathbb{R}^{n_m} \dots \dots \dots \text{Modal velocity states} \\ u &\in \mathbb{R}^m \dots \dots \dots \text{Control input} \\ w_1 &\in \mathbb{R}^p \dots \dots \dots \text{Process noise (disturbance input)} \\ w_2 &\in \mathbb{R}^l \dots \dots \dots \text{Sensor noise} \\ y &\in \mathbb{R}^l \dots \dots \dots \text{Modal acceleration measurements}\end{aligned}$$

For the simply supported plate experiment, $m = p = 1$, because there is only one control shaker and one disturbance shaker. The number of modeled modes, n_m , varies according to the order of the plant for which we wish to design a controller, although it is limited to nine (the total number of modes thus far identified). Since we are modeling n_m modes, the number of modal acceleration measurements is $l = n_m$. This number would be limited to twelve (the number of accelerometers), should there be more than twelve identified modes available. The excess accelerometers provide redundant information about the spatial accelerations from which a more reliable least squares solution to the modal accelerations is computed.

The quantities Ω and Z are diagonal matrices of the modal natural frequencies and damping ratios, respectively. That is,

$$\begin{aligned}\Omega &\triangleq \text{diag}\{\omega_1, \omega_2, \dots, \omega_{n_m}\} \\ &= 2\pi \cdot \text{diag}\{f_1, f_2, \dots, f_{n_m}\}\end{aligned}\tag{6.3}$$

$$Z \triangleq \text{diag}\{\zeta_1, \zeta_2, \dots, \zeta_{n_m}\}$$

where the modal natural frequencies and damping ratios are given in Table 6.1.

Table 6.1: Natural Frequencies and Damping Ratios for the First Nine Modes

| i | f_i (Hz) | ζ_i |
|-----|------------|-------------|
| 1 | 49.447 | 0.007722826 |
| 2 | 108.96 | 0.01171460 |
| 3 | 130.25 | 0.008318498 |
| 4 | 188.53 | 0.002731109 |
| 5 | 203.25 | 0.002725023 |
| 6 | 265.62 | 0.002387554 |
| 7 | 285.78 | 0.001224449 |
| 8 | 326.08 | 0.001321583 |
| 9 | 338.30 | 0.002220888 |

The column vectors Φ_u and Φ_w are the modeshapes corresponding to the control and disturbance inputs, respectively. They indicate the relative effect each shaker has on the modal amplitudes and are a function of the shaker locations. These modeshapes are related to the sensor modeshape matrix, Φ , introduced in Chapter 5 [see (5.11)]. According to the data provided by the identification procedure, the matrix Φ for a model of the first nine modes is given (to four places past the decimal) by:

$$\Phi = \begin{bmatrix} 0.2068 & -0.5350 & 0.4716 & -0.6245 & -0.4200 & -0.3708 & -0.5257 & -0.6133 & -0.2620 \\ 0.4290 & -0.7181 & 0.1104 & -0.1516 & -0.6814 & 0.4496 & -0.0798 & 0.6420 & -0.4180 \\ 0.3072 & -0.4771 & -0.5241 & 0.6138 & -0.4607 & -0.3951 & 0.4640 & -0.6078 & -0.2086 \\ 0.4001 & -0.2333 & 0.8233 & -0.1619 & 0.3468 & -0.6213 & 0.5346 & -0.2604 & 0.4473 \\ 0.7033 & -0.3062 & 0.1353 & -0.0554 & 0.5246 & 0.6389 & 0.1319 & 0.3292 & 0.6070 \\ 0.4829 & -0.1404 & -0.9120 & 0.4227 & 0.3599 & -0.6038 & -0.6515 & -0.2957 & 0.3680 \\ 0.4468 & 0.2215 & 0.7428 & 0.6210 & 0.3438 & -0.6273 & 0.6039 & 0.3454 & -0.3710 \\ 0.7033 & 0.3664 & 0.1091 & 0.2185 & 0.5573 & 0.6715 & 0.1343 & -0.4100 & -0.5521 \\ 0.4480 & 0.2853 & -0.8733 & -0.4771 & 0.3848 & -0.6201 & -0.6411 & 0.3428 & -0.3926 \\ 0.2803 & 0.4538 & 0.5085 & 1.0755 & -0.5294 & -0.3369 & -0.7840 & 0.6357 & 0.1768 \\ 0.4365 & 0.7018 & 0.0766 & 0.2210 & -0.7149 & 0.4368 & -0.1769 & -0.8063 & 0.3551 \\ 0.1884 & 0.4729 & -0.5331 & -1.0680 & -0.4109 & -0.3970 & 0.7980 & 0.5814 & 0.1820 \end{bmatrix} \quad (6.4)$$

The twelve rows of Φ are the modeshapes for the twelve accelerometer locations. The control and disturbance shakers are each colocated with one of the accelerometers, so Φ_u and Φ_w are given by two of the rows of Φ . Denote the i^{th} row of Φ by ϕ_i . Then

$$\Phi_u = \phi_8^T, \quad \Phi_w = \phi_5^T \quad (6.5)$$

The expression for g_{22} was developed in section 5.1 [see (5.14b)]. When fewer than nine modes are modeled (i.e., $n_m < 9$), the appropriate columns of Φ are eliminated. For example, in order to model only the first four modes, we take Φ to be equal to the first four columns of Φ in (6.4).

The control input, u , passes through a second-order smoothing filter before reaching the shaker. If we denote the control command entering the filter by u_c , a state-space model of the smoothing filter is:

$$\begin{aligned} \dot{x}_l(t) &= A_l x_l(t) + B_l u_c(t) \\ u(t) &= C_l x_l(t) \end{aligned} \quad (6.6)$$

where

$$A_l = \begin{bmatrix} 0 & 1 \\ -\omega_l^2 & -2\zeta_l\omega_l \end{bmatrix}, \quad B_l = \begin{bmatrix} 0 \\ \omega_l^2 \end{bmatrix}, \quad C_l = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad (6.7)$$

$$\omega_l = 2\pi \cdot 120, \quad \zeta_l = .707$$

The disturbance w_1 is modeled as narrowband noise centered about 60 Hz. Conceptually, this was accomplished by passing the fictitious white noise variable v through a second-order noise shaping filter. The state-space model of that shaping filter is as follows:

$$\begin{aligned} \dot{x}_w(t) &= A_w x_w(t) + B_w v(t) \\ u(t) &= C_w x_w(t) \end{aligned} \quad (6.8)$$

where

$$A_w = \begin{bmatrix} 0 & 1 \\ -\omega_w^2 & -2\zeta_w\omega_w \end{bmatrix}, \quad B_w = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C_w = \begin{bmatrix} 0 & 1 \end{bmatrix} \quad (6.8)$$

$$\omega_w = 2\pi \cdot 60, \quad \zeta_w = .1$$

Controllers are designed from the augmented plant model, comprised of the interconnected plate, smoothing filter, and noise shaping filter dynamics — (6.1), (6.6), and (6.8). The n^{th} -order augmented plant is described by the following state and output equations:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu_c(t) + G_1 w(t) \\ y(t) &= Cx(t) + G_2 w(t) \end{aligned} \quad (6.9)$$

where

$$x(t) \triangleq \begin{bmatrix} x_p(t) \\ x_l(t) \\ x_w(t) \end{bmatrix}, \quad w(t) \triangleq \begin{bmatrix} v(t) \\ w_2(t) \end{bmatrix} \quad (6.10)$$

$$A = \begin{bmatrix} A_p & B_p C_l & g_{11} C_w \\ 0 & A_l & 0 \\ 0 & 0 & A_w \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ B_l \\ 0 \end{bmatrix}, \quad G_1 = \begin{bmatrix} 0 & g_{12} \\ 0 & 0 \\ B_w & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} C_p & D_p C_l & g_{21} C_w \end{bmatrix}, \quad G_2 = \begin{bmatrix} 0 & g_{22} \end{bmatrix}$$

The noise vector $w(t)$ is assumed to have identity covariance, so the correlations among the individual elements are specified by G_1 and G_2 , and the relative intensities of the process and sensor noise are adjusted by means of ρ_v , which scales g_{22} [see (6.2)]. As stated in section 5.1, $g_{12} = 0$, since the sensor noise (w_2) only affects the output equation. The presence of the smoothing filter eliminates any feedthrough from the controller command signal to the outputs. However, for many of the experiments simulated in this chapter, the smoothing filter was ignored in order to thoroughly study the robustness design methods on a simple model before stepping up to a full scale model. Without the filter, the x_l states are eliminated and $u_c = u$.

Figure 6.1 shows the frequency responses of the plate (for $n_m = 4$), smoothing filter, and noise shaping filter just described. The frequency response of the plate is represented by a plot of its maximum singular value. That enables the magnitude response of the 4×1 transfer function matrix to be expressed by a single curve. The response of the noise shaping filter is scaled so as to be visible in the magnitude range of the plot. This curve also represents the frequency content of the assumed disturbance, although the actual scaling depends on the intensity of that disturbance and is considered a design parameter. The 120 Hz cutoff frequency of the smoothing filter was chosen to allow adequate control of the observed modes (modes 1-4) while minimizing excitation of the higher frequency unobserved modes.

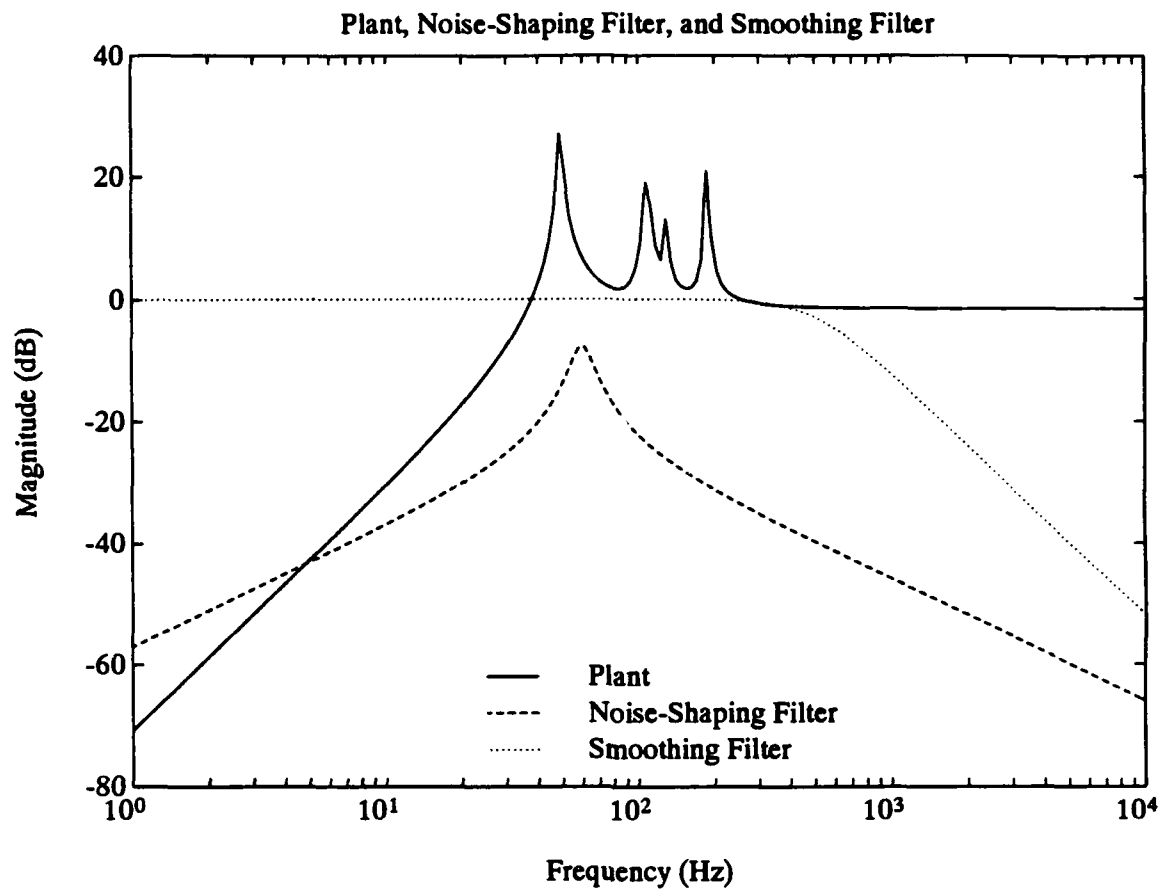


Figure 6.1: Frequency Responses of Augmented Plant Components for 4-Mode Model

6.2 Comparison of Tradeoff for Different Methods

In order to speed up computations and eliminate unnecessary complications, the robust control methods of chapters 2 and 3 were applied to a single mode model of the plate (i.e., $n_m = 1$), and the smoothing filter was omitted from the augmented plant model of (6.9)–(6.10). This continuous-time augmented plant model has order $n = 4$.

An LQG controller was chosen for the baseline design, and after closing the loop it was found that system stability was particularly sensitive to two of the plant parameters — the natural frequency, ω_1 , and the scalar control input eigenvector, Φ_u (corresponding to the control shaker location). The cost functional weighting matrices were chosen by combining the modal energy penalty technique of section 5.1 with the disturbance cancellation algorithm of section 5.2 [using (5.23)], yielding:

$$R_1 = \begin{bmatrix} \omega_1^2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.0002 \end{bmatrix}, \quad R_2 = 1, \quad R_{12} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1.0001 \end{bmatrix} \quad (6.11)$$

Since there is only one modal output, there is no need to worry about relative intensities of multiple sensor noise sources. Therefore, g_{22} was simplified to: $g_{22} = \rho_v^{1/2}$. Then, as usual, the covariance matrices were given by:

$$V_1 = G_1 G_1^T, \quad V_2 = G_2 G_2^T, \quad V_{12} = G_1 G_2^T \quad (6.12)$$

The sensitivity of system stability with respect to ω_1 and Φ_u varied greatly with the selection of ρ_v . For relatively large ρ_v (i.e., relatively small disturbance intensity), stability is more sensitive to ω_1 and less sensitive to Φ_u , whereas for relatively small ρ_v , the reverse is true.

Both cases — sensitivity to ω_1 and Φ_u — were studied in order to test the effectiveness of the various parameter-robustness methods under different conditions. If one method were clearly and consistently superior to the others, it would stand out during these tests. Since the same auxiliary output weighting technique is common to LQG/PRE, the frequency-domain method of Chapter 2, and the time-domain method of

Chapter 3, the comparative effectiveness of these methods was studied by improving robustness of system stability using only auxiliary input noise. Separately, only auxiliary output weighting was used. Then the effect of combining the two was demonstrated.

A. Uncertainty in Natural Frequency

When the cost matrices in (6.11) are combined with the noise intensity parameter, $\rho_v = 10^{-6}$, a +3% error in the natural frequency is enough to drive the closed-loop system (with LQG controller) unstable. In order to evaluate the robust controller designs, a specification was made to raise the stability margin to $\pm 10\%$, and the resulting loss in performance was compared for four different methods:

Auxiliary inputs only:

- (1) White noise (w.n.)
- (2) Frequency-shaped noise (f.s.n.)
- (3) Multiplicative white noise (m.w.n.)

Auxiliary outputs only:

- (4) Auxiliary output penalty (aux. output)

Note that the term "stability margin", as used here and in the remainder of this thesis, does not denote a gain or phase margin in the sense of classical control theory. Rather, it is an abbreviation for *parameter stability margin* and represents the maximum amount a parameter may deviate from its nominal (i.e., assumed) value without causing system instability.

The natural frequency appears in both the A - and C -matrices, so methods (1), (2), and (4) require that we factor both ΔA and ΔC to obtain [see (1.8)]:

$$M_a = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad M_c = 1, \quad N_a = N_c = \begin{bmatrix} 2.05 & 1 & 0 & 0 \end{bmatrix} \quad (6.13)$$

The natural frequency appears in matrix elements $-\omega_1^2$ and $-2\zeta_1\omega_1$ in both A and C , so it is impossible to find a relative scaling for the corresponding elements of ΔA and ΔC that will hold for any size deviation of ω_1 . For a +5% deviation in ω_1

(or $-2\zeta_1\omega_1$), there is a $+2.05 \times 5\%$ deviation in ω_1^2 (or $-\omega_1^2$). This 2.05 ratio very nearly holds over the entire parameter range of interest, hence the values of N_a and N_c in (6.13).

In applying method (2), the noise shaping filter $H_a(s)$ was computed directly from equation (2.11), without a low-order approximation. The full-order shaping filter did not raise the controller order excessively, because the original plant order was so low. Also, due to the structure of the parameter uncertainties, the same noise shaping filter was used for $H_c(s)$, so no additional dynamics were necessary for that filter. Therefore the controller order for method (2) was $n_c = 8$, as opposed to $n_c = 4$ for the other methods.

The design parameters needed to just meet the $\pm 10\%$ stability margin specification are as follows:

- (1) White noise $\mu_a = \mu_c = 400$
- (2) Frequency-shaped noise (scale H_a and H_c by a factor of 1100)
- (3) Multiplicative white noise $\gamma_1 = .64$,

$$A_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 2.05 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$C_1 = [2.05 \ 1 \ 0 \ 0]$$

- (4) Auxiliary output penalty $\rho_a = \rho_c = 600$

Figure 6.2 shows the results of the performance/stability robustness tradeoff for all four robust controllers, compared with the baseline LQG controller. Both frequency-shaped noise and multiplicative white noise provided more suitable auxiliary input models than the white noise of LQG/PRE, although the multiplicative white noise design was clearly the best performer of the three. The auxiliary output penalty design gave up the least performance of all in achieving the stability robustness objective. At the nominal value of ω_1 , that design, and the multiplicative white noise design, are nearly optimal in performance.

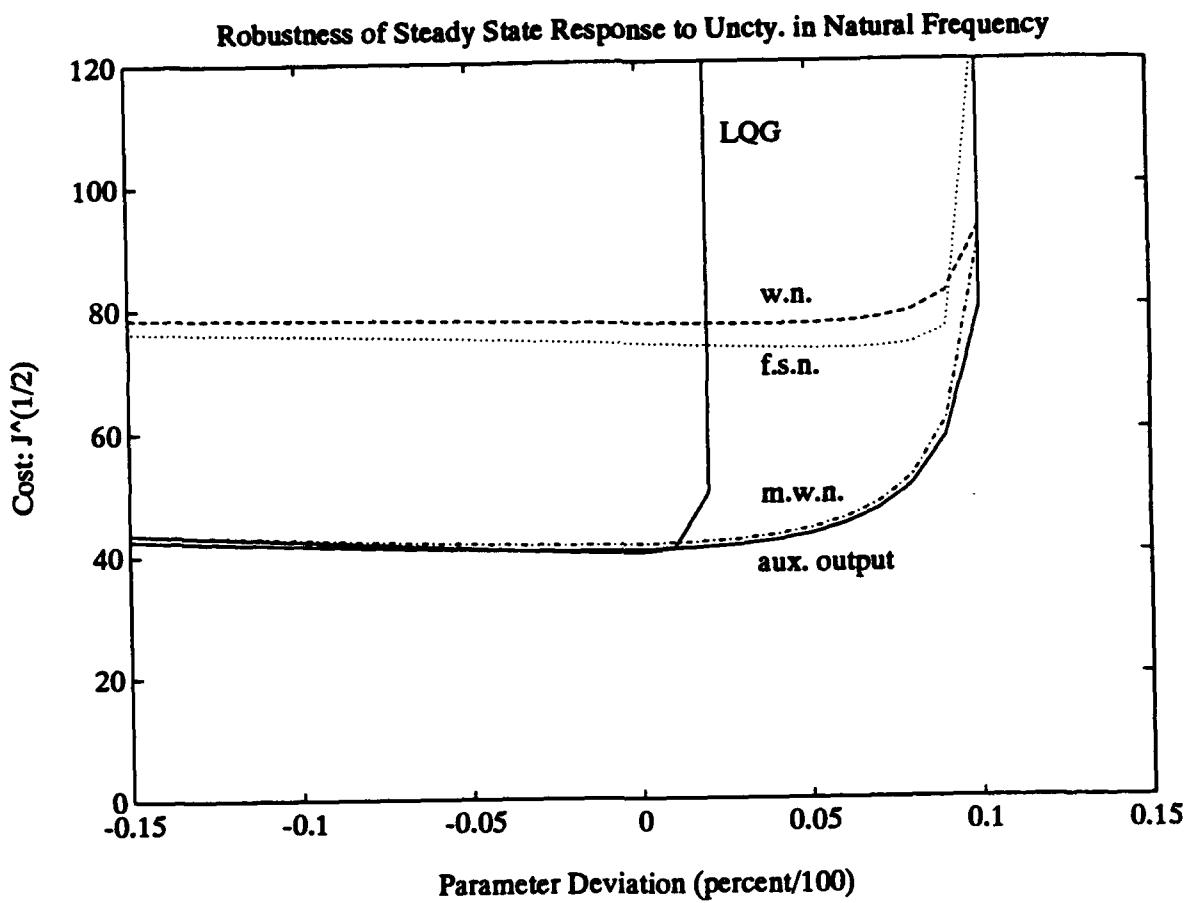


Figure 6.2: Steady State Performance and Stability Tradeoff for Uncertainty in Natural Frequency

The experiment depicted in Figure 6.2 was based on the assumption of steady state process noise — the premise of LQG optimal control. This comparison was repeated with an impulse applied at the process noise port [i.e. the signal v in (6.10)] in order to simulate the effect of transient disturbances on the designs. The results are shown in Figure 6.3. The LQG curve no longer represents optimality at zero parameter deviation, rather it is only an approximation to optimality. In fact, designs (3) and (4) both outperformed the LQG controller in the presence of an impulse disturbance. The multiplicative white noise design performs relatively somewhat better when the disturbance is an impulse, as was expected. The signal levels, and therefore the auxiliary input amplitudes, vary greatly over the course of time, and the multiplicative white noise model takes advantage of this. On the other hand, the finite-energy disturbance input conflicts with the premise of the frequency-shaped noise design, and it performs relatively worse.

B. Uncertainty in Eigenvector

In order to make the closed-loop system sensitive to the control input eigenvector, Φ_u , the sensor noise intensity parameter, ρ_v , was lowered to $\rho_v = 1/(4.9 \times 10^9)$. That resulted in the same baseline stability margin as in the natural frequency uncertainty problem just discussed. A +3% deviation in Φ_u (in the B_p -matrix only) destabilized the closed-loop system with LQG controller. The matrix $D_p = \Phi_u$ was left unaltered for this study, because hardware experimentation showed uncertainty in B_p to be a particular problem.

The control input eigenvector appears in the second row of the augmented B -column vector, so the obvious choice of a factorization for ΔB was:

$$M_b = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad N_b = 1 \quad (6.14)$$

The same stability margin specification was made as before — to improve that margin to $\pm 10\%$. The following design parameters enabled the four robustness methods to meet that specification:

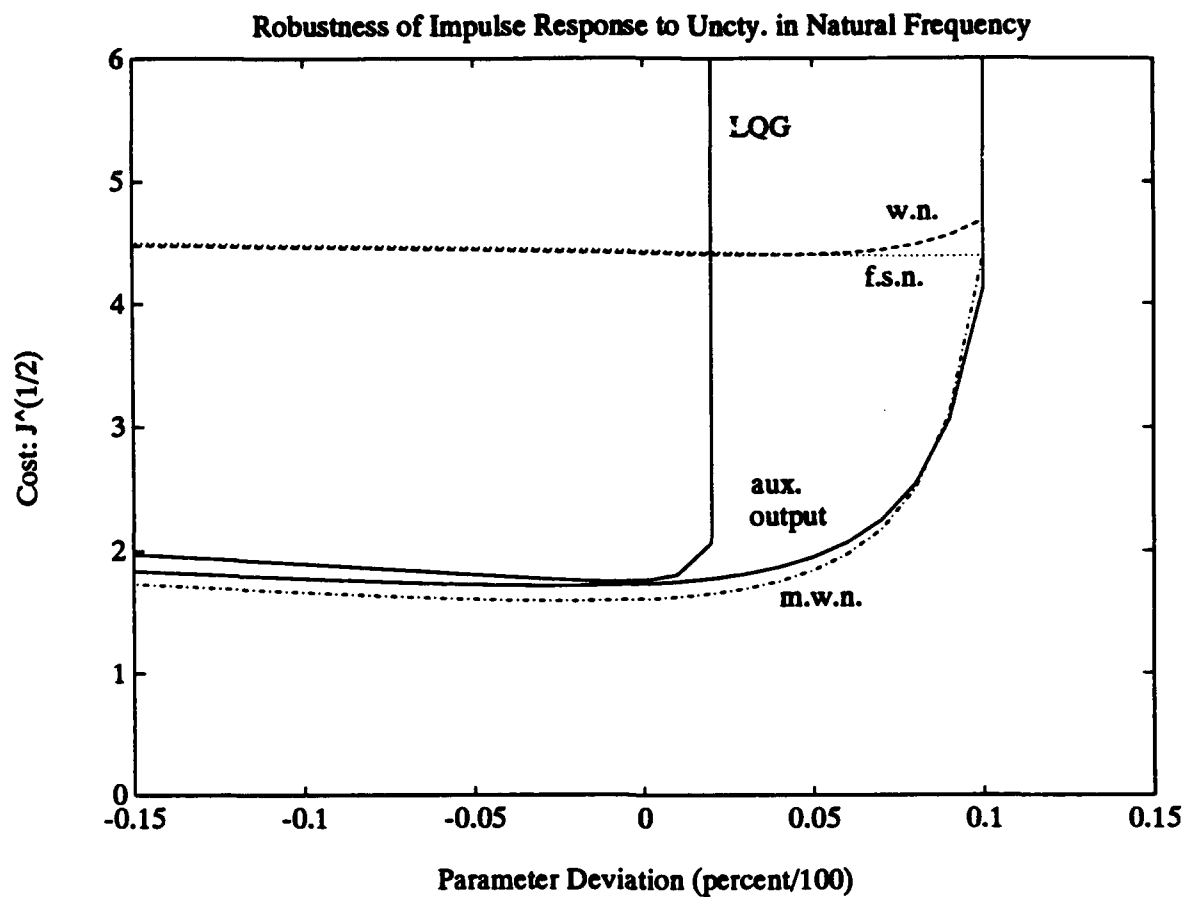


Figure 6.3: Transient Performance and Stability Tradeoff for Uncertainty in Natural Frequency

- (1) White noise $\mu_b = 70$
- (2) Frequency-shaped noise (scale H_b by a factor of .15)
- (3) Multiplicative white noise..... $\gamma_1 = 1.44 \times 10^{-6}$,

$$B_1 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

- (4) Auxiliary output penalty $\rho_b = 3.4$

The steady state disturbance results for this problem are shown in Figure 6.4. The auxiliary output penalty method, rather than being the most effective, was by far the worst performer this time. Among the three auxiliary input models, multiplicative white noise once again gave very good results, but provided no improvement over the simple white noise model. Surprisingly, the frequency-shaped noise model led to a somewhat greater sacrifice in performance than the white noise model.

Plots of the impulse response costs for this example are shown in Figure 6.5. These results are not significantly different from those of the steady state disturbance case.

C. Results for Different Factorizations

Up to now, only the individual components of the parameter robustness methods under study have been compared. A logical procedure for choosing among LQG/PRE, the frequency-domain method of Chapter 2, and the time-domain method of Chapter 3 is to determine the corresponding auxiliary input model that gives the best performance, then to combine that auxiliary input model with the auxiliary output penalty common to all three methods. Once a required stability margin has been specified, the auxiliary input noise intensity (through μ_a , μ_b , and μ_c — assuming we are using LQG/PRE) and the auxiliary output penalty (through ρ_a , ρ_b , and ρ_c) may be applied individually or combined in any number of different proportions to just meet the stability specification. For any single independent parameter uncertainty, adjusting the relative magnitude of the μ - and ρ -scalars is equivalent to choosing different factorizations for ΔA , ΔB , and ΔC , giving the corresponding nonzero elements of either the M - or the N -matrices a relatively greater magnitude. Of course, when more than

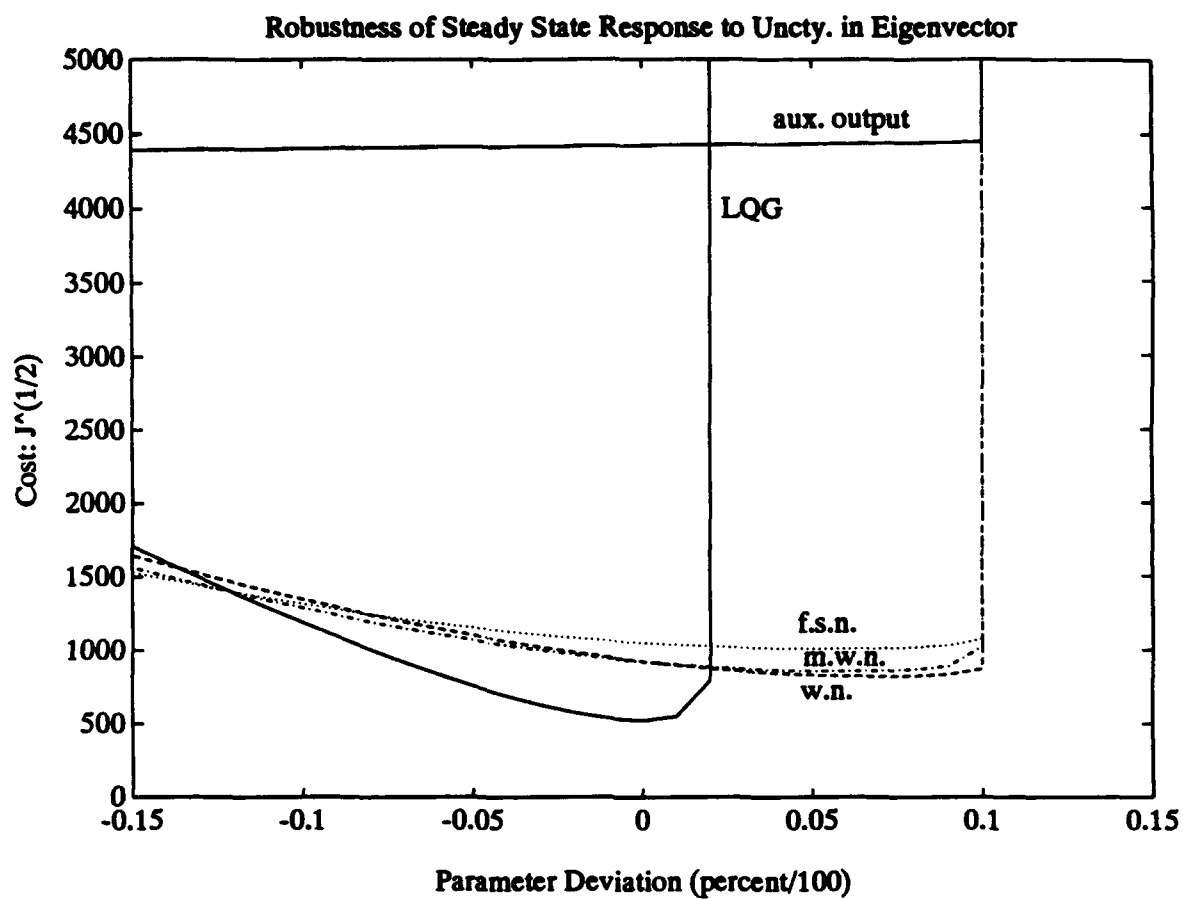


Figure 6.4: Steady State Performance and Stability Tradeoff for Uncertainty in Eigenvector

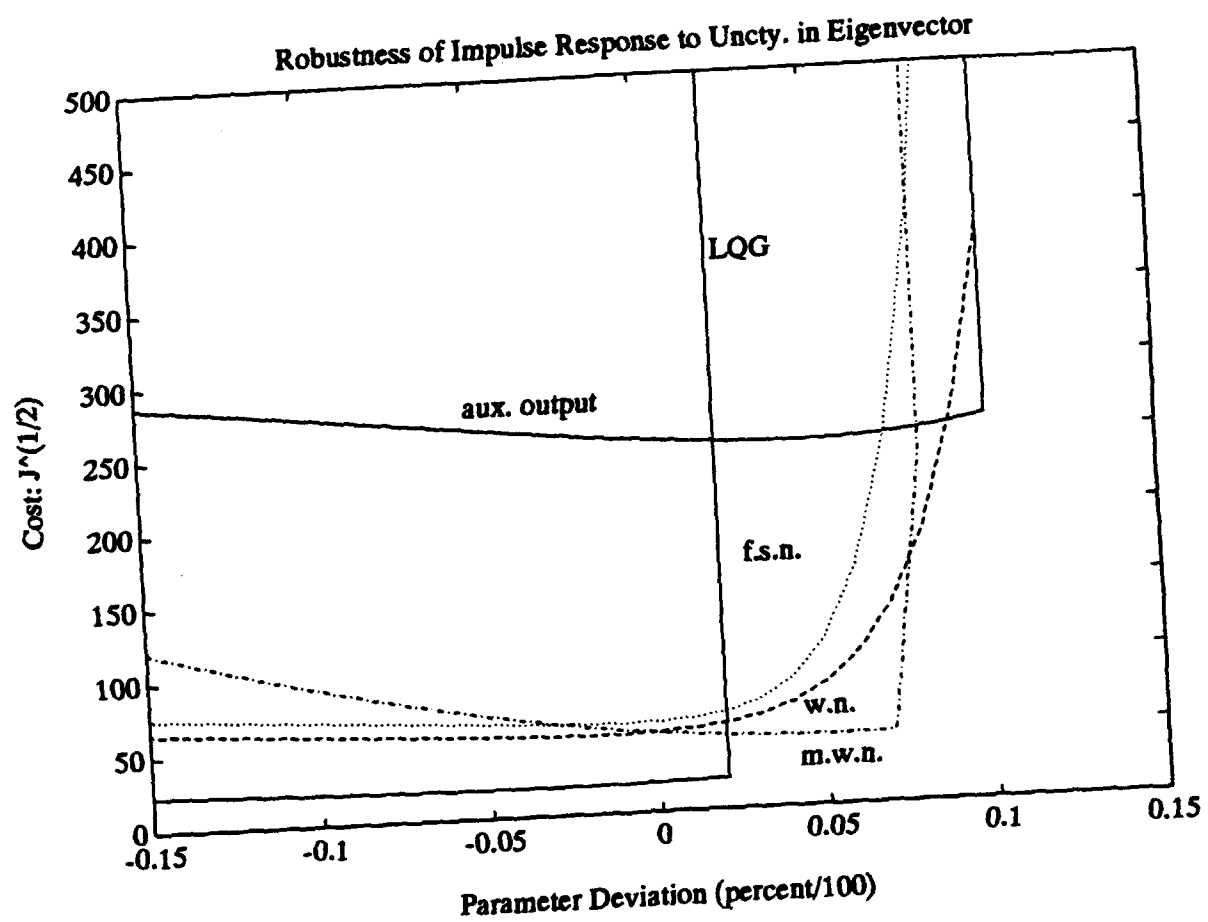


Figure 6.5: Transient Performance and Stability Tradeoff for Uncertainty in Eigenvector

one independent uncertainty is present, the factorizations must be adjusted in order to exercise every degree of freedom available. Analysis of a few different factorizations should reveal one that is very nearly the best possible for any given uncertainty.

This technique was applied to the eigenvector uncertainty problem in section 6.2B. First, the white noise model was chosen for the auxiliary inputs, because of the good performance that results and because of the simplicity of design. Then, holding M_b and N_b constant, the $\pm 10\%$ stability margin was attained using several different combinations of values for μ_b and ρ_b . Those values are detailed in Table 6.2 for eight different controller designs. The performance/stability results of each design are plotted

Table 6.2: Controller Design Parameters Corresponding to Eight Different Factorizations

| | Case Number | | | | | | | |
|----------|-------------|-----|-----|-----|------|-----|------|----|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| μ_b | 0 | 10 | 20 | 30 | 40 | 50 | 60 | 70 |
| ρ_b | 3.4 | 0.9 | 0.5 | 0.3 | 0.15 | 0.1 | 0.05 | 0 |

in Figure 6.6, where the top curve represents Case 1, and the curve for each successive decrease in nominal cost represents the next higher case number. These results are typical of a those from a number of such experiments, including analysis of controllers based on a multiplicative white noise model of the auxiliary inputs. Normally, either the auxiliary input modeling or auxiliary output penalty alone provide virtually the best overall performance possible, although neither one does so consistently. So far, no significant improvement has been achieved by combining auxiliary inputs and outputs.

Blelloch and Mingori (1990) point out that one particular factorization has a certain intuitive value for the problem of multiple natural frequency uncertainties (actually for uncertainties in the $-\omega_i^2$ elements of the A -matrix). Namely, by factoring ΔA such that the nonzero elements of M_a and N_a are equal, the auxiliary outputs lead to a penalty on the elastic strain energy of the flexible structure. However, this approach to factoring ΔA has no bearing on the relative emphasis to be placed on the auxiliary inputs versus the auxiliary outputs. It only decides what relative

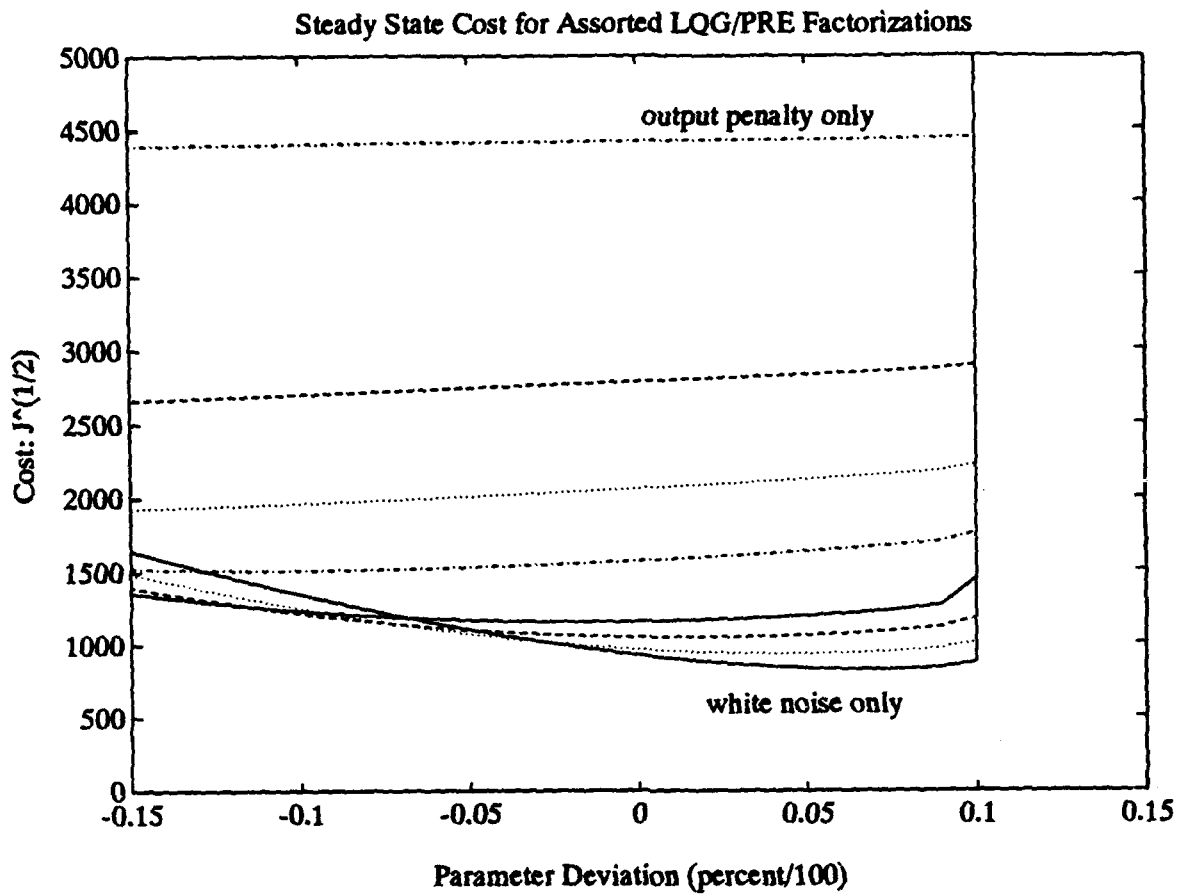


Figure 6.6: Comparative Performance for Several Different LQG/PRE Factorizations

importance should be placed on the various parameter uncertainties. Also, no results were given to draw comparisons with other factorizations.

6.3 Itô vs. Stratonovich Noise

In the examples considered in the previous section, the multiplicative white noise models were interpreted in the sense of Itô. Recall from section 3.5 that in order to interpret multiplicative white noise as Stratonovich noise, we need only to make the following substitutions in the controller design equations:

$$A \leftarrow A + \frac{1}{2} \sum_{i=1}^P \gamma_i A_i^2, \quad (3.73)$$

$$B \leftarrow B + \frac{1}{2} \sum_{i=1}^P \gamma_i A_i B_i,$$

$$C \leftarrow C + \frac{1}{2} \sum_{i=1}^P \gamma_i C_i A_i$$

The Stratonovich noise interpretation was applied to the natural frequency uncertainty problem of section 6.2A (using auxiliary input modeling only) so as to compare the controller performance with that of the Itô noise design already discussed. For a $\pm 10\%$ stability margin requirement, we have the following matrix substitutions for A and C :

$$A \leftarrow A + \frac{1}{2} \gamma_1 A_1^2 \quad (6.15)$$

$$A \leftarrow A + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0.656 & 0.32 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

where on the right-hand side of (6.15):

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -96525 & -4.7987 & 0 & .70334 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -142120 & -75.398 \end{bmatrix}$$

and

$$C \leftarrow C + \frac{1}{2} \gamma_1 C_1 A_1 \quad (6.16)$$

$$C \leftarrow C + \begin{bmatrix} .656 & .32 & 0 & 0 \end{bmatrix}$$

where on the right-hand side of (6.16):

$$C = \begin{bmatrix} -96525 & -4.7987 & 0 & .70334 \end{bmatrix}$$

The stability specification was met with auxiliary input noise of such a low intensity that the modifications to A and C were insignificant. The difference in the stability and performance characteristics of the closed-loop systems for the two noise interpretations was therefore not discernible.

6.4 Reduced-Order Controller

A much higher order model was used to investigate the benefits of optimal reduced-order control. This study employed a 4-mode model of the plate, along with the second-order smoothing filter and second-order disturbance shaping filter to create a twelfth-order augmented plant. The design objectives for the reduced-order controllers were threefold: (1) to provide 15 dB of rejection at the assumed disturbance center frequency (60 Hz), (2) to minimize the increase in the linear quadratic cost over that of the full-order design, and (3) to provide some transient suppression along with the steady state rejection.

The disturbance cancellation method of section 5.2 was found to provide over 15 dB of rejection at 60 Hz for a full-order controller. Combining 60 Hz cancellation with a penalty on the modal energies supplied some transient suppression as well. The cost functional matrices which resulted were similar to those used for the robustness studies:

$$R_1 = \text{diag}\{\Omega^2, I_{4 \times 4}, 0, 0, 0, 1.0002\}, \quad R_2 = 1, \quad R_{12} = \begin{bmatrix} 0_{11 \times 1} \\ 1.0001 \end{bmatrix} \quad (6.17)$$

where the 4×4 natural frequency matrix Ω is defined in (6.3). As before the noise

intensity parameter was chosen to be $\rho_v = 10^{-6}$. The simple model for g_{22} was used: $g_{22} = I$. Then V_1 , V_2 , and V_{12} were computed from (6.12).

A full-order controller was designed for this model, then its performance, frequency response, and time response characteristics were compared with optimal controllers of orders 4, 3, and 2. Table 6.3 quantifies the comparative performance of all four controllers.

Table 6.3: LQ Cost and Disturbance Rejection vs. Controller Order

| Order | 12 | 4 | 3 | 2 |
|-----------|---------|---------|---------|--------|
| Rejection | 15.9 dB | 17.9 dB | 14.8 dB | 5.3 dB |
| LQ Cost | 1576 | 1623 | 1633 | 2753 |

Approximately 15 dB of rejection at 60 Hz is retained for a controller order as low as 3, and very little rejection is lost by substituting this 3rd-order controller for a 12th-order controller. However, the 2nd-order controller is markedly worse in performance. Note how the linear quadratic cost rises with each reduction in controller order. On the transition from a 3rd-order to a 2nd-order controller, the increase in cost is very steep.

Figure 6.7 shows the magnitude frequency responses of the first modal acceleration to the disturbance shaker input for the open-loop system and three closed-loop designs. The 3rd-order controller appears to be the design of choice. We see here how closely the closed-loop system behavior with this controller resembles that of the system with full-order controller. When the controller order is restricted to less than three, however, the optimality (in the full-order sense) of the design begins to break down. All three controllers provide some transient suppression by lowering the open-loop peak at 49 Hz due to the lightly damped first mode at that frequency. However, the 2nd-order controller fails to produce a notch at the 60 Hz disturbance center frequency. By subtracting the open-loop response from each of the closed-loop responses, we arrive at the disturbance rejection plots in Figure 6.8. The vertical line is drawn at 60 Hz to indicate where the 15 dB of rejection is desired. The 2nd-order controller is not able to simultaneously dampen the first mode and provide good rejection at 60 Hz. The frequency response plots of the controllers in Figure 6.9

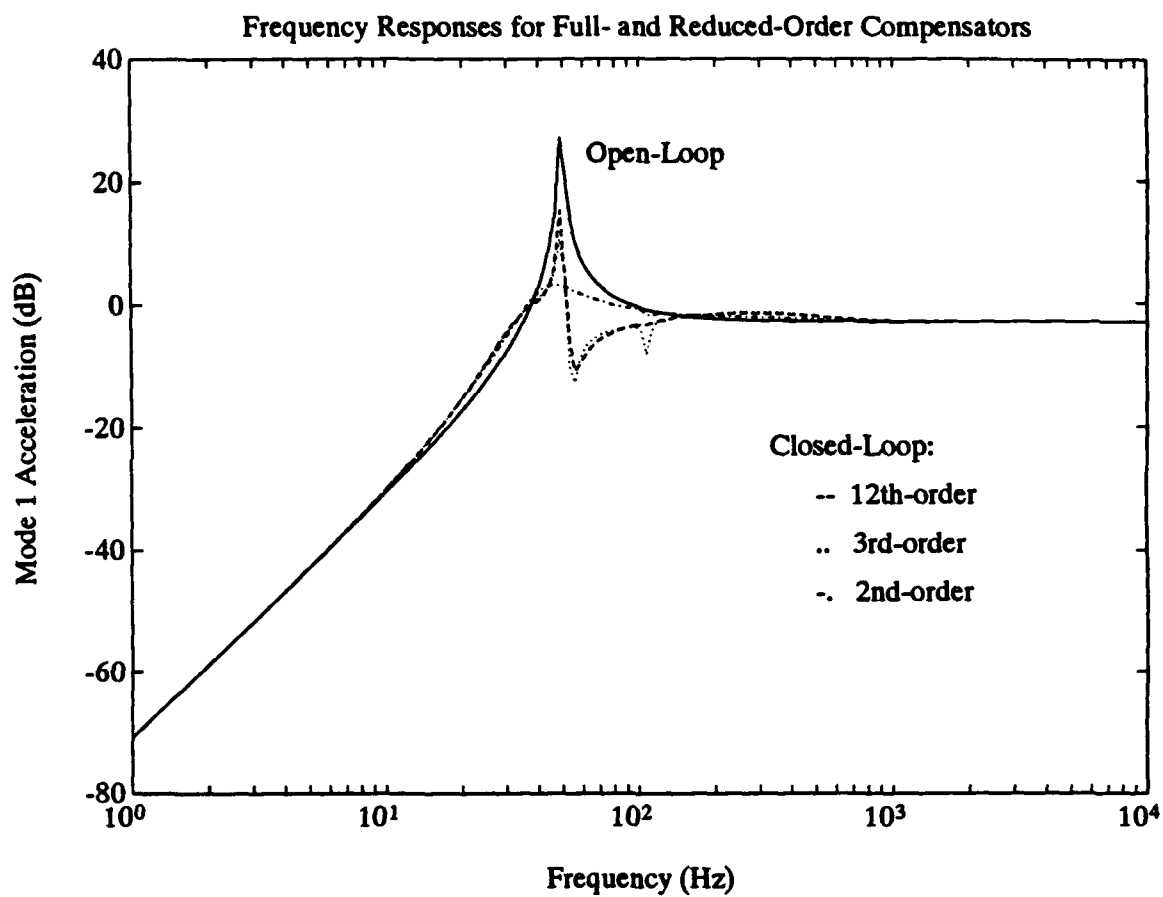


Figure 6.7: Frequency Responses for Full- and Reduced-Order Compensators

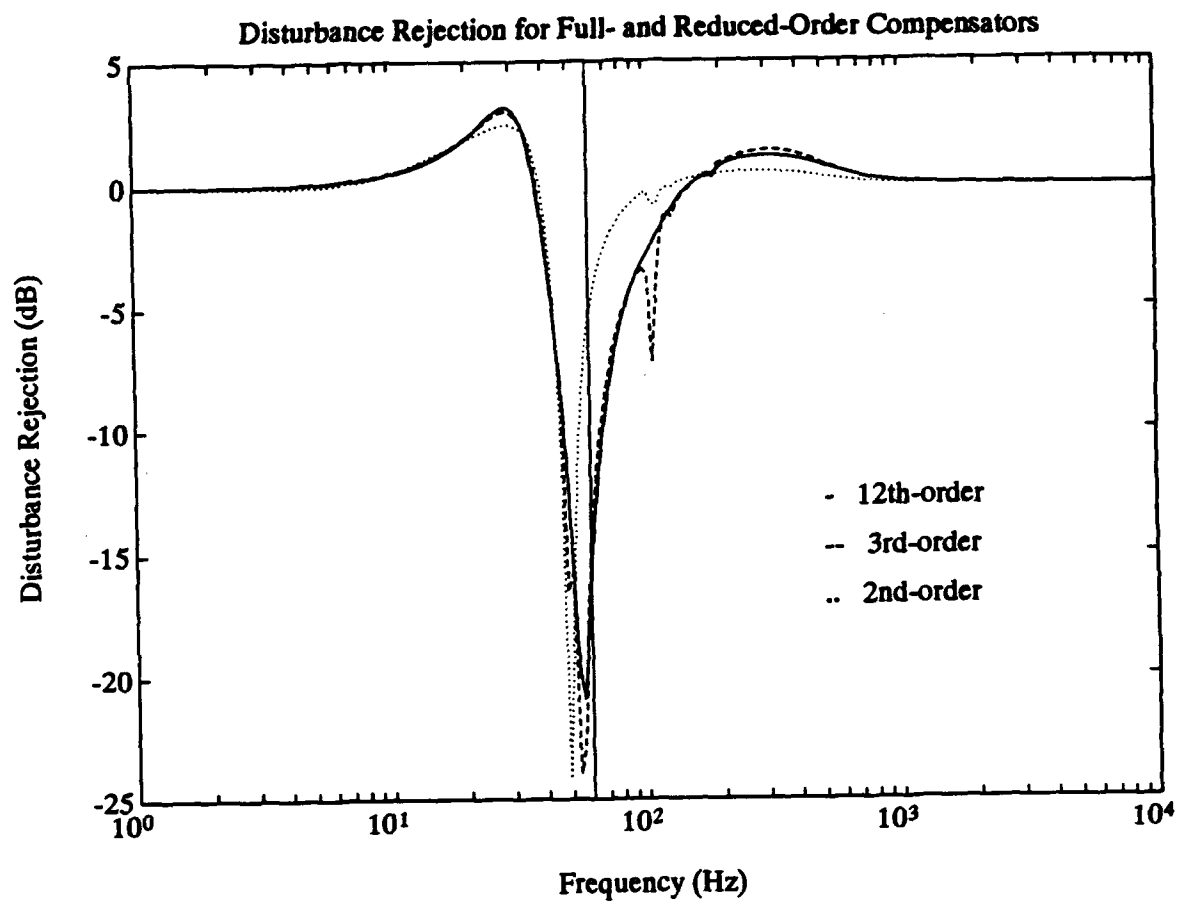


Figure 6.8: Disturbance Rejection for Full- and Reduced-Order Compensators

demonstrate the cause of this limitation. These curves represent the response of the transfer function from the first modal acceleration input to the control output. The response of the 2nd-order controller can approach the full-order response only at the asymptotes. The rapid change of the full-order response in the 49-60 Hz region cannot be matched at the same time by a transfer function with only two poles. In order to compare the time responses of the closed-loop system with 12th- and 3rd-order controllers, a 60 Hz disturbance was applied and the first modal acceleration was measured. These results are shown in Figure 6.10. Again, we see how little performance is sacrificed to achieve this dramatic reduction in controller order.

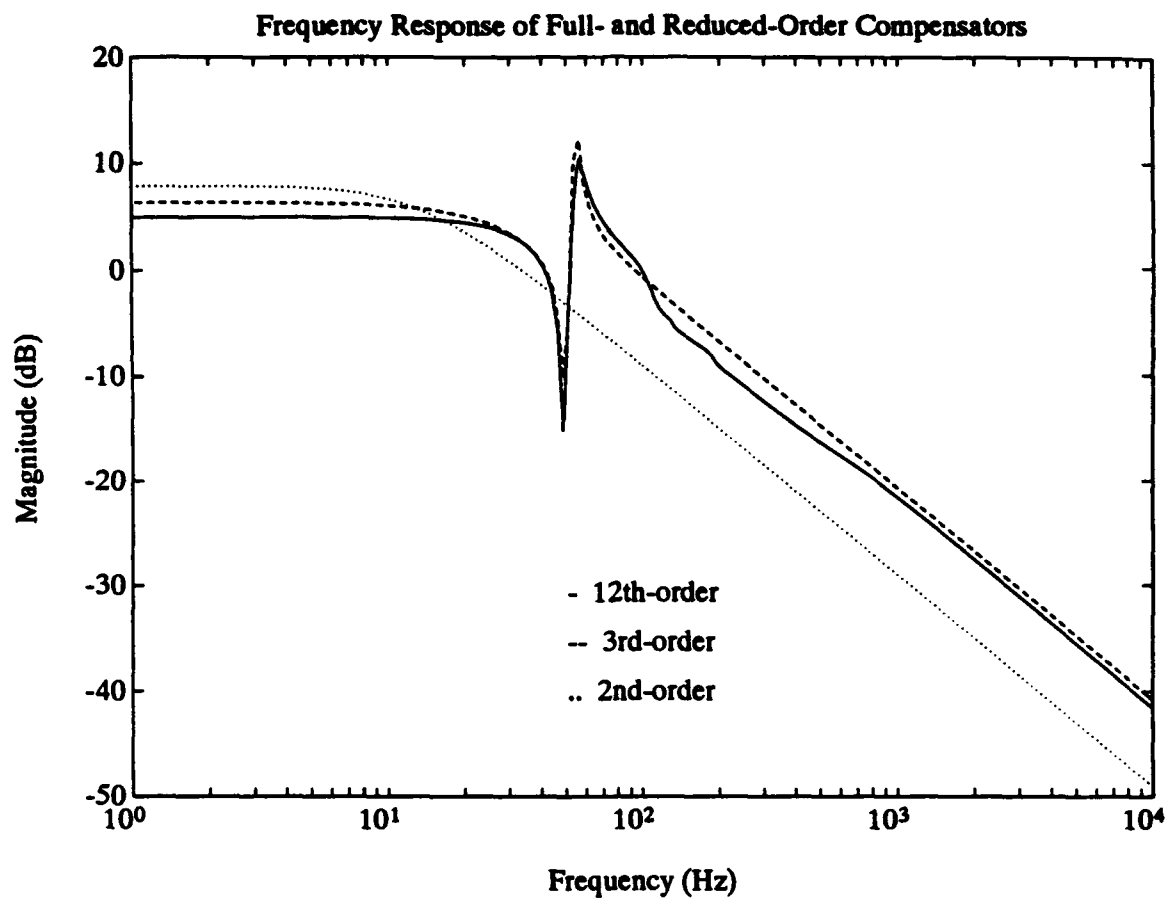


Figure 6.9: Frequency Response of Full- and Reduced-Order Compensators

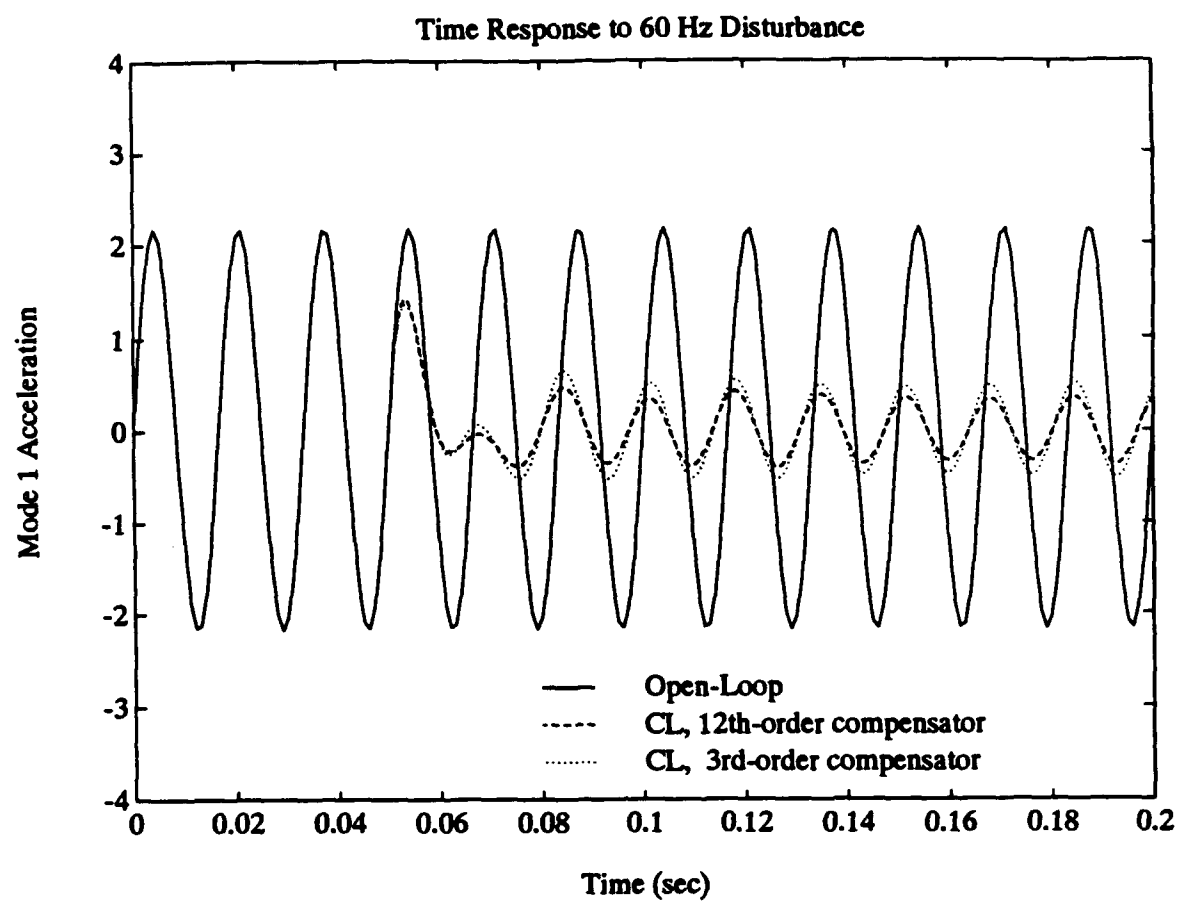


Figure 6.10: Time Response for Full- and Reduced-Order Compensators

7. Application to Simply Supported Plate

7.1 Stability Robustness Problem and its Solution

For experimentation with the simply supported plate hardware, a sinusoidal 60 Hz disturbance was applied at the disturbance shaker. When standard LQG controllers were implemented, very little disturbance rejection could be accomplished. Large controller authorities (in the form of a small penalty, R_2 , on the control input) resulted in system instability due modeling errors of the plate. In order to determine possible sources of the modeling errors, the control penalty was gradually lowered and the LQG controller was repeatedly redesigned until instability resulted. Then the sensitivity of the model to deviations in its parameters was analyzed. The augmented plant model from which the controllers were designed was the same 4-mode, 12th-order model described in section 6.4. The covariance matrices were also the same, with the minor exception that g_{22} was computed as in (6.2), yielding a slightly different value for V_2 . In order to keep matters simple, direct 60 Hz disturbance cancellation was not attempted. Rather, a penalty was applied to the control input and the modal energy of the first mode only. The plant, cost, and covariance matrices were then translated to zero-order hold equivalent form, and the LQG controllers were designed directly in discrete-time. The following (continuous-time) cost functional matrices resulted in instability of the simply supported plate:

$$R_1 = \text{diag}\{\omega_1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0\}, \quad R_2 = 2 \times 10^5, \quad R_{12} = 0 \quad (7.1)$$

Continuous-time analysis of the closed-loop system model with LQG controller designed from these cost matrices revealed that the system was robust to all of the plate's damping ratios and modal frequencies except one. A -5% or a +6% deviation in the second natural frequency was found to destabilize the system model. The nominal natural frequency for the second mode (from Table 6.1) is approximately 109 Hz. Therefore, if the plant model were otherwise accurate and the actual second natural frequency of the plate were anywhere outside the range 103.5–115.5 Hz, instability would result. In order to correct this problem, a requirement was placed on the stability margin with respect to ω_2 — to improve the margin to $\pm 10\%$. Three different modified LQG designs were applied to just meet this specification: (1) white noise auxiliary input modeling, (2) multiplicative white noise input modeling, and

(3) auxiliary output penalty. Since the simultaneous application of auxiliary inputs and outputs was found in Chapter 6 to have little success, no combination of the two was attempted. Also, frequency-shaped auxiliary input noise was not considered, because of its relative ineffectiveness during experimentation with the model and because it would result in a controller of order greater than twelve, meaning that the controller would be too slow for implementation without using some kind of order reduction.

The factorization used to compensate for the natural frequency uncertainty is analogous to the one used in section 6.2A. for the 1-mode model:

$$M_a = \begin{bmatrix} 0_{5 \times 1} \\ 1 \\ 0_{6 \times 1} \end{bmatrix}, \quad M_c = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad N_a = N_c = \begin{bmatrix} 0 & 2.05 & 0_{1 \times 3} & 1 & 0_{1 \times 6} \end{bmatrix} \quad (7.2)$$

The design parameters that the three robust designs used to meet the $\pm 10\%$ stability margin requirement are:

- (1) White noise $\mu_a = \mu_c = .22$
- (2) Multiplicative white noise $\gamma_1 = 2.89$,

$$A_1 = \begin{bmatrix} & & 0_{5 \times 12} & & & & \\ 0 & 2.05 & 0 & 0 & 0 & 1 & 0_{1 \times 6} \\ & & 0_{6 \times 12} & & & & \end{bmatrix},$$

$$C_1 = \begin{bmatrix} & & 0_{1 \times 12} & & & & \\ 0 & 2.05 & 0 & 0 & 0 & 1 & 0_{1 \times 6} \\ & & 0_{2 \times 12} & & & & \end{bmatrix}$$

- (3) Auxiliary output penalty $\rho_a = \rho_c = 7 \times 10^4$

The performance/stability tradeoff results for all three candidate designs is depicted in Figure 7.1, along with the LQG results. Even though the white noise auxiliary input model gave very poor nominal performance in the 1-mode natural frequency uncertainty example of section 6.2A., that same method of robust design sacrificed virtually no nominal performance here and was clearly the best performer overall. The multiplicative white noise design did provide a larger stability margin for positive

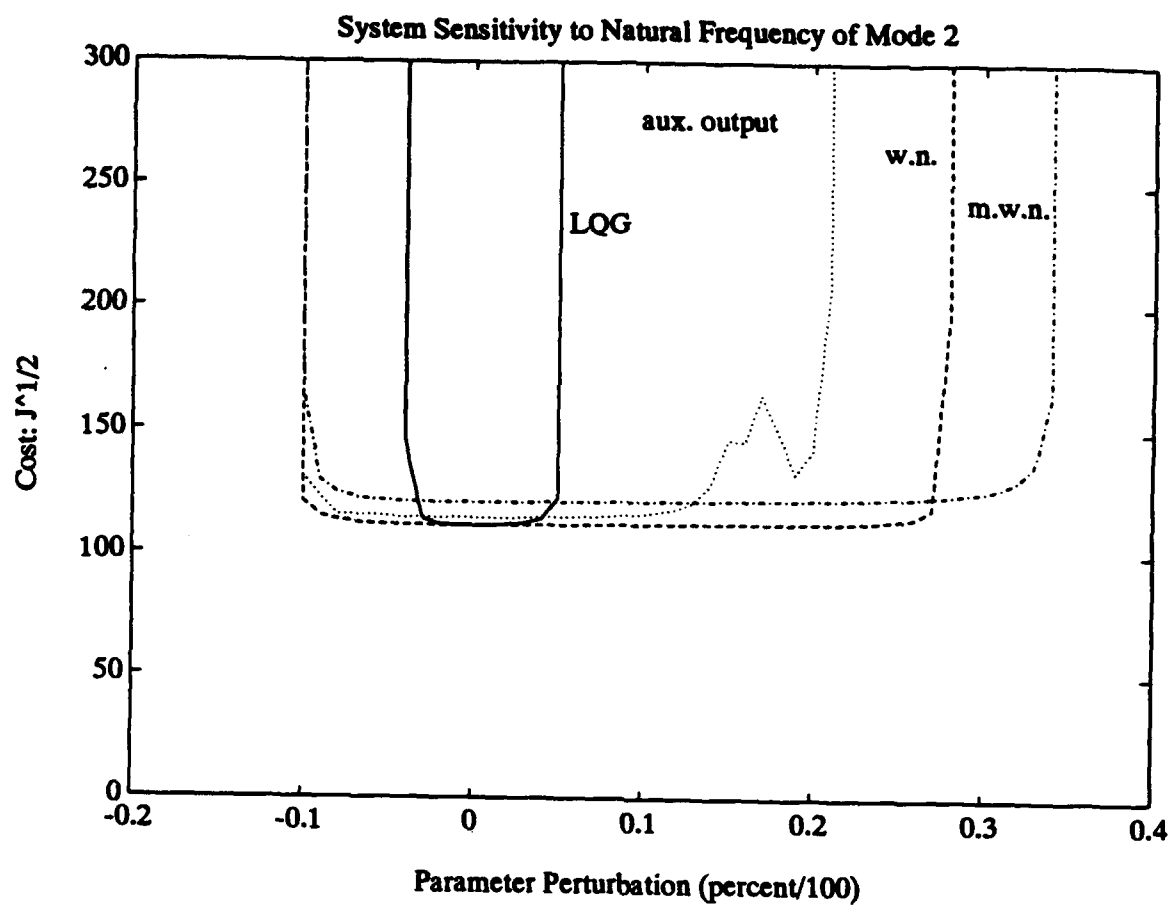


Figure 7.1: Performance/Stability Robustness Tradeoff for Uncertainty in ω_2

deviations in ω_2 , but the natural frequency is surely known to within 28%, so the additional stability margin to +34% is of no value.

The white noise auxiliary input model requires only that we make the following two substitutions for the original covariance matrices:

$$\begin{aligned} V_1 &\leftarrow V_1 + \mu_a M_a M_a^T \\ V_2 &\leftarrow V_2 + \mu_c M_c M_c^T \end{aligned} \quad (7.3)$$

Then we use LQG to arrive at a different Kalman filter (but the same regulator gains). This parameter robust controller design was implemented in the hardware and did in fact stabilize the system. The measured time responses of mode 1 and mode 2 accelerations to the 60 Hz disturbance are shown in Figures 7.2 and 7.3, respectively. After the loop was closed, the LQG controller rejected the response in mode 1, but only temporarily, as mode 2 was driven unstable. The robust controller stabilized mode 2 (and the entire system) while giving up only a small amount of rejection in mode 1.

7.2 Effectiveness of Reduced-Order Controllers

Robust optimal controllers of orders 3 and 4 were designed using the same white noise auxiliary input model and the same cost and covariance matrices used to design the full-order robust controller discussed above. For this level of controller authority, the iterative relaxation method of section 5.4 proved to be sufficient for solving the optimal projection equations. Since a sampled-data version of the optimal projection equations was not available, the reduced-order controllers were designed in continuous-time, then discretized by means of a bilinear transformation. The 3rd- and 4th-order controllers significantly reduced the computational delay caused by the full-order controller, so the one-sample-interval delay was ignored on the first attempt. Then, in each case, a second controller was designed based on a 2nd-order Padé approximation of the delay. This time delay model raised the order of the augmented plant from 12 to 14.

The sample rates for all three controller orders are given in Table 7.1.

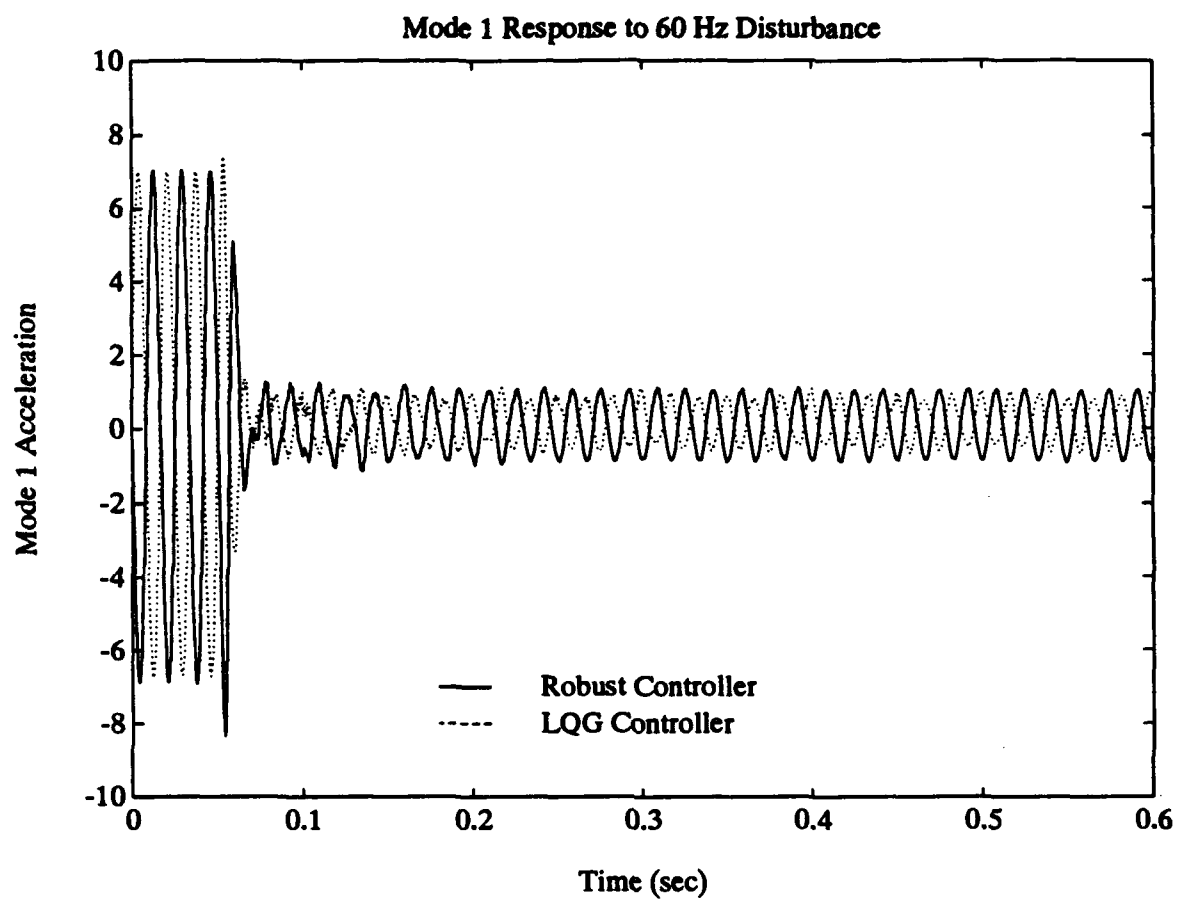


Figure 7.2: Robust Control — Mode 1 Response to 60 Hz Disturbance

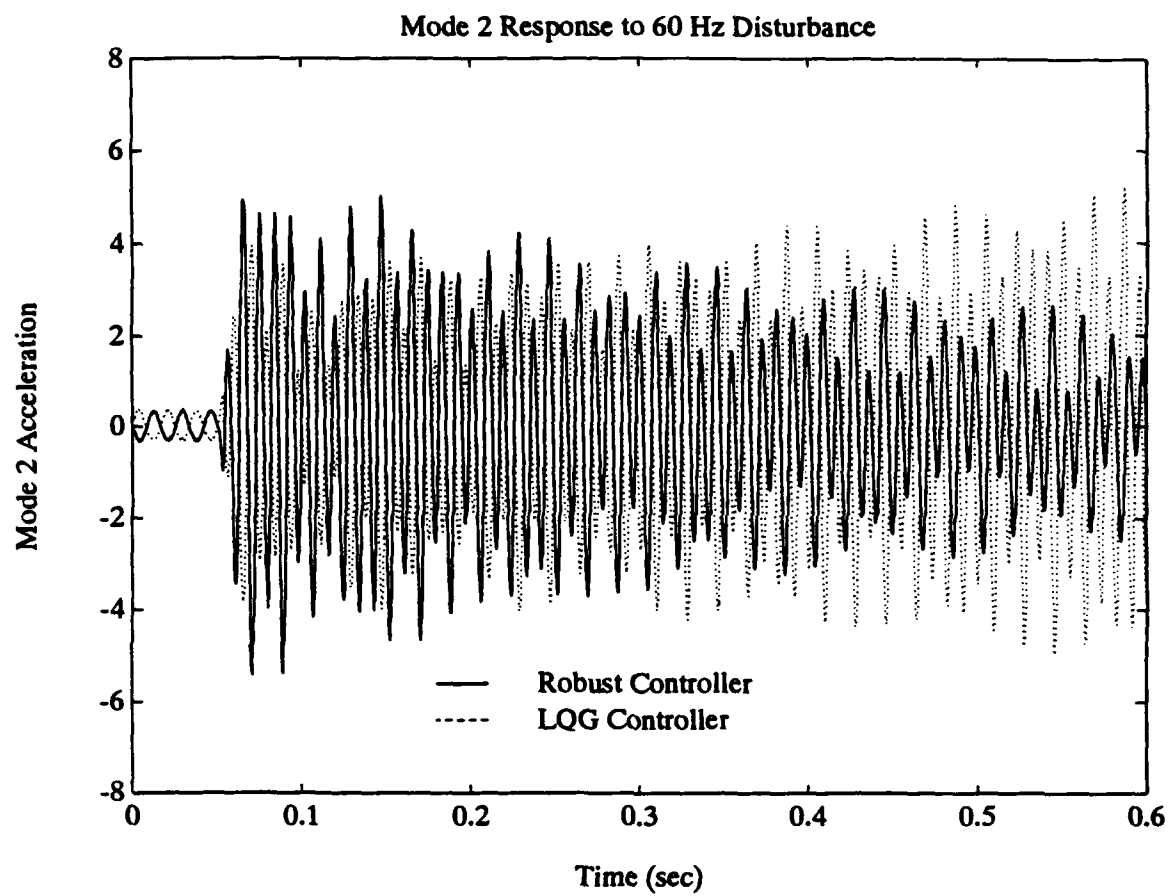


Figure 7.3: Robust Control — Mode 2 Response to 60 Hz Disturbance

Table 7.1: Sample Rates for Full- and Reduced-Order Controllers

| Controller Order | 12 | 4 | 3 |
|------------------|------|------|------|
| Sample Rate (Hz) | 1997 | 3615 | 4023 |

The improvement of the reduced-order sample rates over that of the full-order design was not as dramatic as one might expect, because the full-order controller was able to take advantage of the structure of the modal state-space realization of the plant. It exploited the sparseness of the block-diagonal system matrix and the sparseness of the output matrix to speed up the Kalman filter considerably.

In order to examine the effect of computational time delay on control system performance, an unmodeled delay of one sample interval was introduced into the full-order robust controller by preventing the Kalman filter from predicting one step ahead. The delay did not cause instability, but controller performance suffered somewhat. Figure 7.4 shows the comparative responses of the first mode to a 60 Hz disturbance for full-order controllers with and without the time delay.

Even though continuous-time analysis predicted the 3rd-order controller would stabilize the system, it did not, regardless of whether the computational delay was modeled. The 4th-order controller stabilized the system only when the delay was modeled, and in that case gave quite good performance. Again, a 60 Hz disturbance was applied to the plate, and the responses of modes 1 and 2 were measured. In Figures 7.5 and 7.6, these results are compared with those of the full-order controller. The 4th-order controller does give up a small amount of disturbance rejection, but increases the sample rate significantly. The significance of this experiment is its demonstration that parameter robust and reduced-order control can be accomplished simultaneously to stabilize a flexible structure and reject disturbances with minimal computational delay.

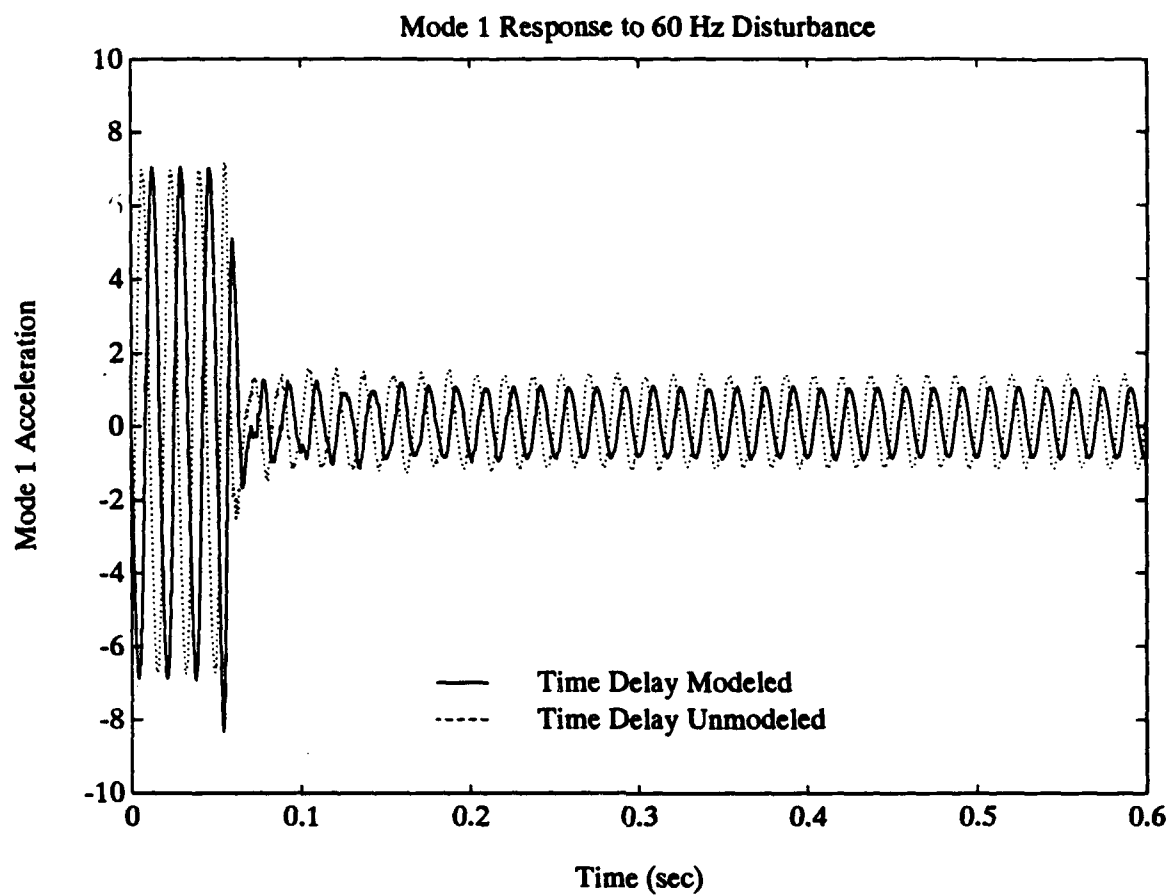


Figure 7.4: Response of Full-Order Robust Controllers With and Without Time Delay

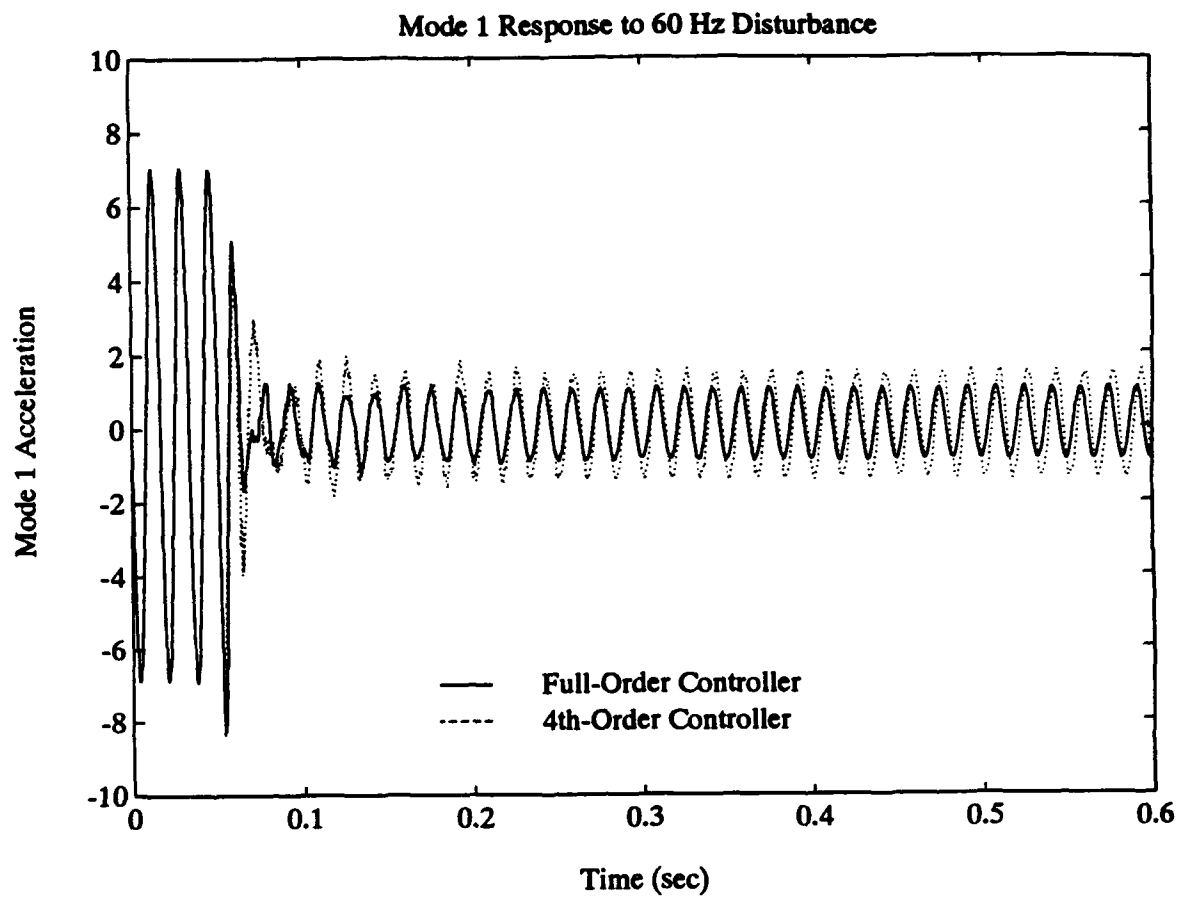


Figure 7.5: Robust Reduced-Order Control — Mode 1 Response to 60 Hz Disturbance

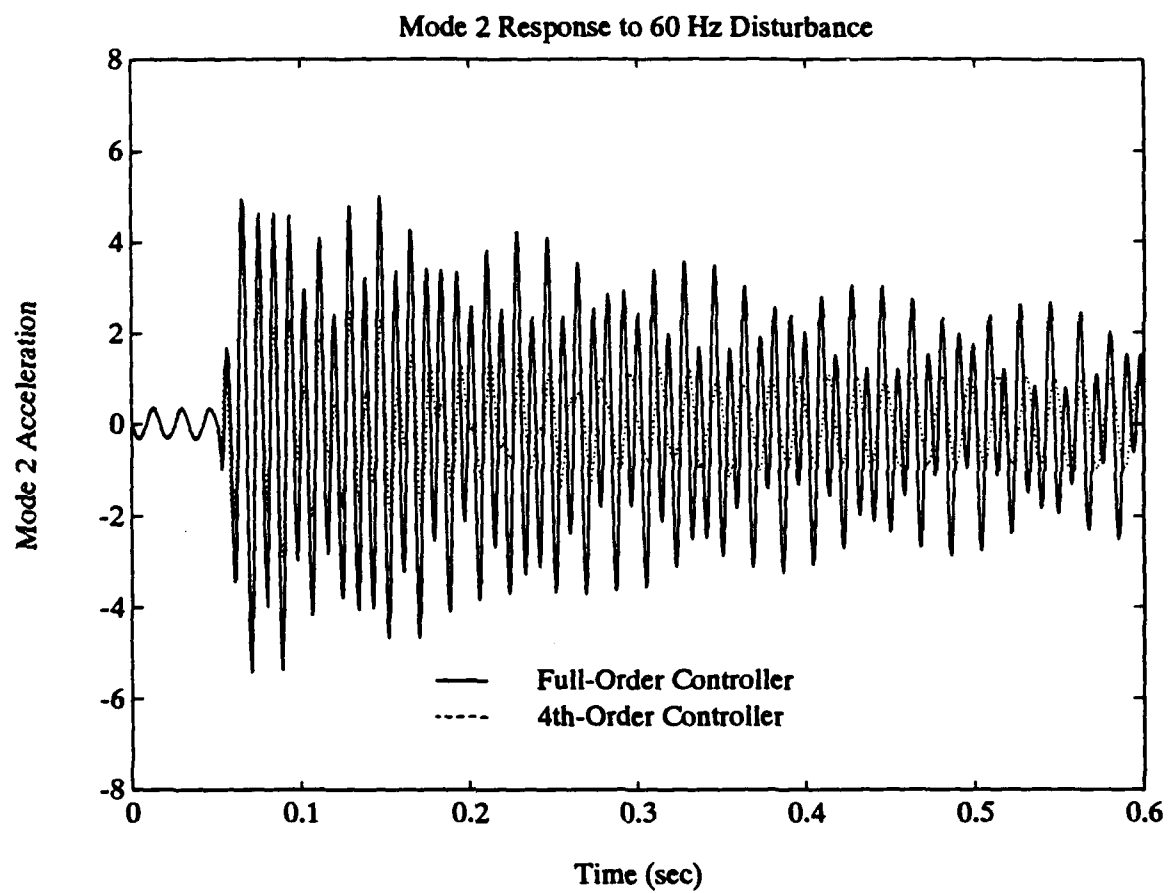


Figure 7.6: Robust Reduced-Order Control — Mode 2 Response to 60 Hz Disturbance

8. Conclusions

As stated in the introduction, the objectives of this work were to compare a number of different parameter robustness techniques based on the LQG/PRE error model, to investigate the limitations on controller order reduction, and to evaluate the combined robust minimal order design.

8.1 Robustness

The few parameter robust control problems considered in Chapters 6 and 7 serve as a counterexample to the proposition that any one of the robust design techniques studied is consistently superior to the others. In fact, the white noise, multiplicative white noise, and auxiliary output designs each provided the best performance for at least one problem. The best performer in one case was found to be the worst performer, by far, in another. Also, the frequency-shaped noise design proved to be a contender at times. Therefore, no strong conclusions can be drawn on the basis of performance alone.

Although the combination of an auxiliary input and auxiliary output model provides greater flexibility in the design, experience seems to indicate that only rarely is any advantage gained from such a combination. Even then, the advantage appears to be very slight. This result is useful, since the elimination of this flexibility from the design allows us to eliminate the complexity of choosing among an infinite number of *MLN*-factorizations for each parameter uncertainty. Excellent performance and stability characteristics are attained consistently by choosing the better of the two extremes — auxiliary inputs only or auxiliary outputs only. Assuming this simplification, we may discuss the comparative merits of the time-domain method, the frequency-domain method, and LQG/PRE in complete generality by considering the three auxiliary input models and the auxiliary output model each on its own.

The multiplicative white noise model for the auxiliary inputs was the most consistent of the four models in providing good performance for a specified stability margin. Relative to the other models, this one tends to show added improvement when the signal amplitudes of the plant vary greatly over time, such as is the case when the plant is subjected to transient disturbances. However, the performance provided by this model was repeatedly matched or beaten by simpler models.

The frequency-shaped noise model has not proven to be superior for any example studied thus far. Although it has in some cases resulted in a controller with better performance than one designed using a white noise model, the added design complexity would seldom justify the small advantage this model might provide. The intent of frequency shaping is to model the low and high frequency rolloff of the auxiliary input frequency response. However, the frequency content of white noise outside the passband of the system (which poorly models reality) is greatly attenuated and apparently has no significant harmful effect. In the presence of transient disturbances, the frequency-shaped noise model tends to impair performance, since its design relies on the frequency spectrum of an assumed steady state disturbance.

The white noise auxiliary input model and the auxiliary output penalty each yield poor performance in some cases. However, one or the other consistently provides very good performance, and the simplicity of their associated controller design methods has important advantages. Both models require only the modification of cost or covariance matrices in a standard LQG design. Therefore, they do not require the iterative design computations of the multiplicative white noise model, nor do they increase controller order, as does the frequency-shaped noise model. The rapidity of controller design allows more effort to be devoted to analysis, which is crucial when there are multiple independent parameter uncertainties.

8.2 Controller Order

For a 12th-order augmented plant (14th-order with the computational delay model) we were able to design a 3rd-order controller in theory and a 4th-order controller in practice without sacrificing a great deal of performance. Instead of determining the minimal controller order that is practical by successively designing optimal controllers of many different orders, we may examine the frequency response of the full-order controller to find out what order a controller must have to display the significant features of that response. That order serves as a good first iteration to the minimum controller order desirable. The addition of more modes to the model would not tend to raise the order of a controller necessary to provide a good response, because any additional modeled modes would necessarily be less dominant. This effect was demonstrated by the addition of the 2nd-order computational delay model to the plant

in Chapter 7. Not only did the minimum controller order fail to rise, it fell due to the more precise modeling of the plant.

8.3 Parameter Robust Reduced-Order Design

The application of the optimal projection equations to the auxiliary input and output models of LQG/PRE is new and proved to be very successful. Its application to the LQG/PRE error model in general makes these methods of parameter robust controller design much more powerful. The reduction in computational delay makes active vibration control possible when it otherwise may not have been. Also, optimal reduced-order design allows us to model more dynamics to give the controller a more accurate picture of the plant, thus reducing another source of uncertainty — unmodeled dynamics. The additional dynamics may include more disturbance modeling, smoothing filter modeling, and time delay modeling. In the robust reduced-order design of Chapter 7, the sacrifice in performance may have been reduced or eliminated by giving up some of the improvement in sample rate in favor of modeling more high frequency modes. This experiment was not carried out, because it would have required rewiring of the smoothing filter, which was designed for a 4-mode model. However, the 4th-order robust controller as designed did stabilize the system and provide a significant amount of disturbance rejection.

8.4 Directions for Further Study

The most conspicuous shortcoming of the design methods and algorithms presented in this study is the lack of a method of designing optimal reduced-order and multiplicative white noise based controllers directly in discrete-time. This deficiency results in the introduction of two additional sources of error. Firstly, the design of an optimal continuous-time controller followed by the transformation of that controller into a discrete-time equivalent is a suboptimal process. Secondly, for the continuous-time design an approximation is necessary to create a finite-dimensional model of the inevitable computational delay. In a continuous-time augmented plant model, the computational delay is not only imperfectly modeled, it may also require the addition of high order dynamics in order to be modeled adequately, particularly when the plant is multi-input multi-output. The solution of the optimal reduced-order control problem

for sampled-data systems is of particular interest, since it would have widespread application in LQG-based design.

References

- Anderson, Brian D. O., and Yi Liu. 1989. Controller reduction: Concepts and approaches. *IEEE Transactions on Automatic Control* 34 (August): 802-812.
- Arnold, Ludwig. 1974. *Stochastic Differential Equations: Theory and Applications*. New York: John Wiley & Sons.
- Balas, Gary J., and John C. Doyle. 1990. Identification of flexible structures for robust control. *IEEE Control Systems Magazine* 10 (June): 51-58.
- Bender, Carl M., and Steven A. Orszag. 1978. *Advanced Mathematical Methods for Scientists and Engineers*. New York: McGraw-Hill.
- Bernstein, Dennis S. 1987. Robust static and dynamic output-feedback stabilization: Deterministic and stochastic perspectives. *IEEE Transactions on Automatic Control* 32 (December): 1076-84.
- Bernstein, Dennis S., and Scott W. Greeley. 1986. Robust controller synthesis using the maximum entropy design equations. *IEEE Transactions on Automatic Control* 31 (April): 362-4.
- Bernstein, Dennis S., and Wassim M. Haddad. 1989. LQG control with an \mathcal{H}_∞ performance bound: A Riccati equation approach. *IEEE Transactions on Automatic Control* 34 (March): 293-305.
- Bernstein, Dennis S., and David C. Hyland. 1988a. Optimal projection equations for reduced-order modelling, estimation, and control of linear systems with multiplicative white noise. *SIAM Journal of Optimization Theory and Applications* 58 (September): 387-409.
- Bernstein, Dennis S., and David C. Hyland. 1988b. Optimal projection for uncertain systems (OPUS): A unified theory of reduced-order, robust control design. In *Large Space Structures: Dynamics and Control*, edited by S. N. Atluri and A. K. Amos, 263-302. Berlin: Springer-Verlag.
- Blelloch, P. A., and D. L. Mingori. 1990. Robust Linear Quadratic Gaussian Control for Flexible Structures. *Journal of Guidance, Control, and Dynamics* 13 (January-February): 66-72.
- Dailey, R. Lane. 1990. Lecture notes. *Workshop on \mathcal{H}_∞ and μ Methods for Robust Control*, San Diego, CA.
- Doyle, John C. 1984. Lecture notes in advances in multivariable control. *ONR/Honeywell Workshop*, Minneapolis, MN.

- Doyle, John C. 1985. Structured uncertainty in control system design. *Proceedings of the 24th Conference on Decision and Control*. Ft. Lauderdale, FL. 260-65.
- Doyle, John C., Kathryn Lenz, and Andy Packard. 1987. Design examples using μ -synthesis: Space shuttle lateral axis fcs during reentry. In *Modelling, Robustness and Sensitivity Reduction in Control Systems*, edited by Ruth F. Curtain, 127-54. Berlin: Springer-Verlag.
- Doyle, John C., and Gunter Stein. 1981. Multivariable feedback design: Concepts for a classical/modern synthesis. *IEEE Transactions on Automatic Control* 26 (February): 4-16.
- Ewing, George M. 1985. *Calculus of Variations with Applications*. New York: Dover.
- Gelb, Arthur (editor). 1984. *Applied Optimal Estimation*. Cambridge, MA: M.I.T. Press.
- Gikhman, I. I., and A. V. Skorokhod. 1969. *Introduction to the Theory of Random Process*. Philadelphia: W. B. Saunders Company.
- Glover, Keith. 1984. All optimal Hankel norm approximations of linear multivariable systems and their L_∞ error bounds. *International Journal of Control*. 39 (June): 1115-93.
- Graham, Alexander. 1981. *Kronecker Products and Matrix Calculus: with Applications*. New York: John Wiley & Sons, Inc.
- Gupta, Narendra K. 1980. Frequency-shaped cost functionals: Extension of linear-quadratic-Gaussian design methods. *Journal of Guidance, Control, and Dynamics* 3 (November-December): 529-35.
- Hyland, David C. 1982. Minimum information stochastic modelling of linear systems with a class of parameter uncertainties. *Proceedings of the American Control Conference*. 620-27.
- Hyland, David C., and Dennis S. Bernstein. 1984. The optimal projection equations for fixed-order dynamic compensation. *IEEE Transactions on Automatic Control* 29 (November): 1034-37.
- Hyland, David C., and Appasaheb N. Madiwale. 1981. A stochastic design approach for full-order compensation of structural systems with uncertain parameters. *Proceedings of the AIAA Guidance and Control Conference*. Albuquerque, NM. 324-32.
- Itô, K. 1944. Stochastic integral. *Proc. Imp. Acad. Tokyo* 20: 519-24.
- Jazwinski, Andrew H. 1970. *Stochastic Processes and Filtering Theory*. New York: Academic Press.

- Kalman, R. E. 1964. When is a linear control system optimal? *Journal of Basic Engineering* 86 (March): 51-60.
- Kreindler, Eliezer, and Antony Jameson. 1972. Optimality of linear control systems. *IEEE Transactions on Automatic Control* 17 (June): 349-51.
- Kwakernaak, Huibert, and Raphael Sivan. 1972. *Linear Optimal Control Systems*. New York: John Wiley & Sons.
- Laub, Alan J. 1980. Computation of "balancing" transformations. *Proceedings of the 1980 Joint Automatic Control Conference*. Vol. 2, FA8-E.
- Lin, Jong-Yin. 1989. *Robust Design of LQG Controllers*. Ph.D. Dissertation. University of California, Los Angeles, CA.
- Maciejowski, J. M. 1989. *Multivariable Feedback Design*. Wokingham, England: Addison-Wesley.
- Moore, B. C. 1981. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Transactions on Automatic Control* 26 (January): 17-32.
- Richter, Stephen. 1987. A homotopy algorithm for solving the optimal projection equations for fixed-order dynamic compensation: Existence, convergence and global optimality. *Proceedings of the American Control Conference*. 1527-31.
- Richter, Stephen, and Emmanuel G. Collins, Jr. 1989. A homotopy algorithm for reduced order compensator design using the optimal projection equations. *Proceedings of the 28th Conference on Decision and Control*. Tampa, FL. 506-11.
- Richter, Stephen L., and Raymond A. DeCarlo. 1983. Continuation methods: Theory and applications. *IEEE Transactions on Automatic Control* 28 (June): 660-65.
- Rubenstein, Stephen P. 1991. *An Experiment in State-Space Vibration Control of Steady Disturbances on a Simply Supported Plate*. Ph.D. Dissertation. Virginia Polytechnic Institute and State University, Blacksburg, VA.
- Sievers, Lisa A., and Andreas H. von Flotow. 1989. Comparison of two LQG-based methods for disturbance rejection. *Proceedings of the 28th Conference on Decision and Control*. Tampa, FL. 483-85.
- Stengel, Robert F. 1986. *Stochastic Optimal Control: Theory and Application*. New York: John Wiley & Sons.
- Stratonovich, R. L. 1966. A new representation for stochastic integrals and equations. *J. SIAM Control* 4: 362-71.

- Tahk, Minjea, and Jason L. Speyer. 1987. Modeling of parameter variations and asymptotic LQG synthesis. *IEEE Transactions on Automatic Control* 32 (September): 793-801.
- Tahk, Minjea, and Jason L. Speyer. 1989. Parameter robust linear-quadratic-Gaussian design synthesis with flexible structure control applications. *Journal of Guidance, Control, and Dynamics* 12 (July-August): 460-68.
- Wong, Eugene, and Moshe Zakai. 1965. On the relation between ordinary and stochastic differential equations. *International Journal of Engineering Science* 3: 213-29.
- Wonham, W. Murray. 1967. Optimal stationary control of a linear system with state-dependent noise. *SIAM Journal on Control*. 5: 486-500.
- Wonham, W. Murray. 1968. On a matrix Riccati equation of stochastic control. *SIAM Journal on Control*. 6: 681-697.
- Wonham, W. Murray. 1970. Random differential equations in control theory. In *Probabilistic Methods in Applied Mathematics*, Vol. 2, edited by A. T. Bharucha-Reid, 131-212. New York: Academic Press.

TABLE OF CONTENTS

| | |
|--|----|
| 1.0 INTRODUCTION | 1 |
| 2.0 PREVIOUS RESEARCH AND RELATED WORK | 6 |
| 2.1 Gaussian Sum Estimation | 6 |
| 2.2 Adaptive Kalman Filtering | 9 |
| 2.3 Alternate NonGaussian Filtering | 12 |
| 3.0 NONGAUSSIAN ESTIMATION | 14 |
| 3.1 Gaussian Sum Estimation Technique | 14 |
| 3.2 NonGaussian Signal Model Development | 20 |
| 3.3 Modified Gaussian Sum Adaptive Filtering | 25 |
| 3.4 Simulation Example | 40 |
| 3.5 Comparison of Gaussian Sum and Modified Gaussian Sum Algorithms | 44 |
| 4.0 APPLICATION TO A MODAL SYSTEM | 50 |
| 4.1 Modal System and Filter Development | 50 |

| | | |
|------------|---|------------|
| 4.2 | Simulation Example | 59 |
| 4.3 | Necessary Condition for Effective Estimation | 74 |
| 4.4 | Alternate Estimation Procedures | 82 |
| 4.5 | Comparison of the Alternate Estimation Procedures | 89 |
| 5.0 | PARAMETER UPDATING METHODS | 107 |
| 5.1 | Zero-Bias Measurement Residual Method | 108 |
| 5.2 | Gaussian Double-Sum Method | 116 |
| 6.0 | CONCLUSIONS | 125 |
| | APPENDIX A | 128 |
| | LIST OF REFERENCES | 132 |
| | VITA | 138 |

LIST OF ILLUSTRATIONS

| | |
|--|----|
| Figure 1. Gamma and Gaussian sum probability density functions. | 24 |
| Figure 2. Modified Gaussian sum adaptive filter structure. | 39 |
| Figure 3. (a) Measurement and state simulation, (b) modified Gaussian sum estimate compared with state, (c) error = state - estimate. | 42 |
| Figure 4. (a) Weight 1, (b) weight 2, (c) weight 3, (d) overall measurement bias estimate. | 43 |
| Figure 5. Modified Gaussian sum adaptive filter structure for a modal system. | 58 |
| Figure 6. (a) Stochastic FM signal $u(t)$, (b) probability density function of $u(t)$, (c) normalized histogram of $u(t)$ | 61 |
| Figure 7. Stochastic FM signal and Gaussian sum probability density functions. | 63 |
| Figure 8. Stochastic FM signal and shaping filter power spectral densities. ... | 65 |
| Figure 9. (a) State 1 and MGS adaptive filter estimate, (b) state 1 and augmented Kalman filter estimate. | 70 |
| Figure 10. (a) State 2 and MGS adaptive filter estimate, (b) state 2 and augmented Kalman filter estimate. | 71 |
| Figure 11. (a) Weighting terms, (b) weighting terms compared with stochastic FM signal. | 73 |

| | |
|--|-----|
| Figure 12. Stochastic FM input signal. | 94 |
| Figure 13. (a) Mode 1 acceleration measurement z_k , (b) mode 1 actual velocity measurement y_k and <i>approximate</i> -velocity measurement \tilde{y}_k | 96 |
| Figure 14. Mode 1 position: (a) State and estimate \hat{x}_{k+1}^v , (b) state and estimate $\hat{x}_{k+1}^{\tilde{a}}$, (c) state and estimate $\hat{x}_{k+1}^{\tilde{v}}$, (d) state and estimate $\hat{x}_{k+1}^{\tilde{\zeta}}$ | 98 |
| Figure 15. Mode 1 velocity: (a) State and estimate \hat{x}_{k+1}^v , (b) state and estimate $\hat{x}_{k+1}^{\tilde{a}}$, (c) state and estimate $\hat{x}_{k+1}^{\tilde{v}}$, (d) state and estimate $\hat{x}_{k+1}^{\tilde{\zeta}}$ | 99 |
| Figure 16. Mode 2 position: (a) State and estimate \hat{x}_{k+1}^v , (b) state and estimate $\hat{x}_{k+1}^{\tilde{a}}$, (c) state and estimate $\hat{x}_{k+1}^{\tilde{v}}$, (d) state and estimate $\hat{x}_{k+1}^{\tilde{\zeta}}$ | 100 |
| Figure 17. Mode 2 velocity: (a) State and estimate \hat{x}_{k+1}^v , (b) state and estimate $\hat{x}_{k+1}^{\tilde{a}}$, (c) state and estimate $\hat{x}_{k+1}^{\tilde{v}}$, (d) state and estimate $\hat{x}_{k+1}^{\tilde{\zeta}}$ | 101 |
| Figure 18. Mode 3 position: (a) State and estimate \hat{x}_{k+1}^v , (b) state and estimate $\hat{x}_{k+1}^{\tilde{a}}$, (c) state and estimate $\hat{x}_{k+1}^{\tilde{v}}$, (d) state and estimate $\hat{x}_{k+1}^{\tilde{\zeta}}$ | 102 |
| Figure 19. Mode 3 velocity: (a) State and estimate \hat{x}_{k+1}^v , (b) state and estimate $\hat{x}_{k+1}^{\tilde{a}}$, (c) state and estimate $\hat{x}_{k+1}^{\tilde{v}}$, (d) state and estimate $\hat{x}_{k+1}^{\tilde{\zeta}}$ | 103 |
| Figure 20. (a) Weighting terms, estimator <i>a</i> , (b) weighting terms, estimators <i>b</i> and <i>c</i> , (c) weighting terms, estimator <i>d</i> | 104 |
| Figure 21. (a) Residuals using the zero dc gain force to acceleration modal system, (b) residuals using the nonzero dc gain force to velocity modal system. | 106 |
| Figure 22. Stochastic FM signal and processed zero-bias measurement residual. | 113 |

| | |
|---|-----|
| Figure 23. (a) State and state estimate with no parameter updating, (b) state and state estimate using zero-bias measurement residual parameter updating method. | 114 |
| Figure 24. (a) Weighting terms with no parameter updating, (b) weighting terms using zero-bias measurement residual parameter updating method. | 115 |
| Figure 25. Modified Gaussian double-sum adaptive filter structure. | 121 |
| Figure 26. (a) State and state estimate with no parameter updating, (b) state and state estimate using Gaussian double-sum parameter updating method. | 123 |
| Figure 27. (a) Weighting terms 1-3 with $A_j = 2$, (b) weighting terms 4-6 with $A_j = 5$ | 124 |

LIST OF TABLES

| | | |
|----------|--|-----|
| Table 1. | Gaussian sum P_i , b_i values, $\sigma_n = 1$. | 23 |
| Table 2. | Modified Gaussian sum adaptive filter equations. | 38 |
| Table 3. | Modified Gaussian sum adaptive filter equations for a modal system. | 57 |
| Table 4. | Gaussian sum P_i , b_i , and S_i values. | 62 |
| Table 5. | Normalized mean-square-error percentage for state estimates. | 72 |
| Table 6. | Overall state estimates to be compared. | 90 |
| Table 7. | Parameter values for the three mode model. | 91 |
| Table 8. | Normalized mean-square-error percentage for state estimates. | 105 |
| Table 9. | Modified Gaussian double-sum adaptive filter equations for a modal system. | 120 |

1.0 INTRODUCTION

This investigation is concerned with effective state estimation of a system driven by an unknown nonGaussian input with additive white Gaussian noise, and observed by measurements containing feedthrough of the same nonGaussian input and corrupted by additional white Gaussian noise. The motivation behind this work is the problem of robust modal control of large, multiple input multiple output (MIMO), heavily damped structures. The scope of the problem involves system identification of the system parameters, optimal estimation of the system states, and robust control of the heavily damped modes of the system. This investigation will deal specifically with the state estimation portion of the problem.

In an earlier study [1], Kalman filter estimation techniques [2], [3] are developed for a lightly damped, simply supported plate. The input forcing function is assumed to be narrowband Gaussian, which is adequately modeled by passing white Gaussian noise through a linear, time-invariant filter. Thus, knowledge of the power spectral density of the narrowband process is sufficient for the design of an adequate estimator. Since the plant and measurement equations for this system are linear, and the inputs and plant and measurement disturbances are Gaussian, the Kalman filter will provide the optimum minimum mean-square error estimate

of the system states [4].

Although the Gaussian assumption for modeling many types of inputs and noise processes is valid in a wide range of applications, in practice it may not be a good assumption for some signal models. For instance, also considered in [1] are large, heavily damped structures modeled by a plant with complex modes. The input to the plant is unknown, but its frequency characteristics are known and the input is always present. The signal model chosen to represent this input is a stochastic FM signal, generated by frequency modulating a sinusoid with a Gaussian process. This stochastic FM signal has a highly nonGaussian probability density function. The power spectral density of a signal of this type appears similar to that of a Gaussian narrowband signal. An estimator design based solely on the power spectral density of a nonGaussian signal assumed to be a Gaussian narrowband signal may provide very poor estimates.

Estimation techniques need to be developed that can cope effectively with the nonGaussian nature of certain signals. One such approach is the Gaussian sum technique, developed by Alspach [6-8]. The density function of each nonGaussian process of the system is approximated by a weighted sum of Gaussian density functions. The conditional density of the state given the available measurement sequence, necessary in the Kalman filter development, is updated using the Gaussian sum approximations and Bayes' rule.

A serious limitation in Alspach's approach is that the number of Gaussian terms used to approximate the density functions increases at each time iteration. A modified approach is required to alleviate this limitation. A modified estimation structure is developed based on an adaptive Kalman filter scheme first presented by Magill [16], and extended by Moose [23]. Essentially, a parameter vector is used to uniquely describe each Gaussian term in the estimator. The parameter vector is

restricted to be randomly chosen from the same finite set of known values at each iteration. By using a nonGaussian signal model in conjunction with the modified formulation of the Gaussian sum estimator, the number of Gaussian terms at each iteration of the estimator will be fixed, thereby avoiding the growing memory problem.

The modified estimation algorithm is termed the *modified Gaussian sum adaptive filter* and forms the basis for the nonGaussian state estimation in this investigation. An algorithm similar to this one that does not make reference to the Gaussian sum characteristic is known in the literature as the parallel processing algorithm [4], or the multiple model algorithm [5]. Both the Gaussian sum and modified Gaussian sum algorithms have been implemented in a variety of important engineering applications such as phase and frequency estimation [36], geophysical field navigation [37], maneuvering target tracking [24], and specification of route widths for air traffic controllers [20].

Several contributions to the field of applied estimation theory are made from this investigation. These include:

1. The development of a modified Gaussian sum algorithm with nongrowing memory based on a nonGaussian signal model with a Gaussian sum probability density function. Parameters from this model are used directly in the modified Gaussian sum adaptive filter structure.
2. A comparison between the Gaussian sum (GS) filter of [6] and the modified Gaussian sum (MGS) adaptive filter. The two are similar, but the comparison shows the MGS adaptive filter to be a good improvement to the GS filter.

3. An examination of a necessary condition for effective MGS estimation. This condition provides a simple test to determine if the MGS adaptive filter will work properly for a given system.
4. An alternate configuration of the MGS adaptive filter when the necessary condition of 3 above is not met. This configuration is applied in several ways, and each is evaluated on a performance basis.
5. Two methods of monitoring and updating key parameters of the MGS adaptive filter. These allow the estimator to react to changes in the input signal level which cause the signal to be nonstationary over long periods of time.

The dissertation begins with a literature review in Chapter 2. This chapter covers previous work and applications in the areas of Gaussian sum estimation, adaptive Kalman filtering, and nonGaussian estimation. Chapter 3 begins with the development of the Gaussian sum estimator. Then, a detailed description of the nonGaussian signal model and the alternate formulation of the Gaussian sum density approximation is given. A curve fitting procedure used to find the initial modified Gaussian sum parameter vector is outlined. The MGS adaptive filter is developed for a general system and a simulation example is given. Several differences between the MGS and GS algorithms are examined. Chapter 4 describes the modal system, the nonGaussian input model, and the MGS adaptive filter based on this system. A simulation example using the MGS adaptive filter is given and the results are compared to those produced from an augmented Kalman filter based on an augmented system model assuming a narrowband Gaussian input

signal. A necessary condition for effective MGS estimation is derived. Alternate estimation procedures are developed to compensate for situations when this condition is not met. Several configurations are simulated and their performance results are analyzed and compared. Chapter 5 discusses two methods of monitoring and updating key parameters of the MGS adaptive filter. Simulation results are analyzed to investigate the performance of these methods. Finally, Chapter 6 gives the conclusions and outlines suggested directions for future investigations.

2.0 PREVIOUS RESEARCH AND RELATED WORK

This chapter presents a review of the literature relevant to this investigation. It begins with a discussion of the Gaussian sum estimation technique. Many examples of its use in a wide range of applications are given. Next, an adaptive Kalman filtering method is outlined along with several modifications. Several examples of the various applications of this method are presented. Finally, an overview of alternate nonGaussian filtering and its applications is discussed.

2.1 Gaussian Sum Estimation

An estimation technique, applicable to both linear systems with nonGaussian inputs and nonlinear systems with Gaussian inputs, has been developed by Alspach [6-8]. In the case of linear systems, the noise processes associated with the plant and measurement are assumed known and nonGaussian. The probability density function of each noise process is approximated by a Gaussian sum; that is, a weighted sum of Gaussian density functions. The Gaussian sum approximation is written as

$$p_{GS}(x) = \sum_{i=1}^M \alpha_i N[\mu_i, \sigma_i^2] \quad (2.1.1)$$

where

$$\sum_{i=1}^M \alpha_i = 1; \quad \alpha_i \geq 0 \quad \text{for } i = 1, 2, \dots, M \quad (2.1.2)$$

and

$$N[\mu_i, \sigma_i^2] = \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{1}{2}\left(\frac{x - \mu_i}{\sigma_i}\right)^2} \quad (2.1.3)$$

For sufficiently large M , any density function can be closely approximated by a Gaussian sum. As long as condition (2.1.2) holds, the Gaussian sum is always a valid density function.

The basic filtering problem is to estimate the state x_k of the system from the current measurement sequence $Z_k = \{z_1, z_2, \dots, z_{k-1}, z_k\}$. The "best" estimate \hat{x}_k is found by minimizing or maximizing a particular performance criterion, such as minimizing the mean-square error between x_k and \hat{x}_k . In the Bayesian approach to estimation, this requires the use of the *a posteriori*, or conditional, density function $p(x_k | Z_k)$.

The Gaussian sum estimator [6-8] is briefly described here and is more fully developed in chapter 3. The density functions of the initial state, the plant noise process, and measurement noise process are approximated by Gaussian sum densities of the form of (2.1.1). These densities are then directly used to form the conditional density function $p(x_k | Z_k)$ as a sum of Gaussian densities. An unfortunate consequence of using these Gaussian sum densities is that the number of Gaussian terms that forms $p(x_k | Z_k)$ increases at each stage of the estimator.

This *growing memory* problem is a serious limitation in Alspach's development. In chapter 3, a nonGaussian signal model is developed and used in conjunction with a modified Gaussian sum adaptive filter that avoids the growing memory problem.

Alspach has applied the Gaussian sum technique to a wide variety of applications. In conjunction with Sorenson, he applies the technique to linear systems with nonGaussian noise [7], and to nonlinear systems with Gaussian noise [8]. In [9], [10], and with Scharf and Abiri in [11], he addresses the problem of linear systems with Gaussian noise processes having unknown covariances. Joint identification, tracking, and prediction in a multi-target, multi-sensor environment is considered in [13], and with LaGrotta in [12]. Alspach and Sorenson [14] use the Gaussian sum technique in conjunction with proving the validity of the separation theorem for linear, nonGaussian, optimal control problems. Alspach extends this work in [15] to nonlinear systems. Here he does not use the separation theorem, but certainty equivalence control instead, producing a suboptimal control algorithm useful for off-line investigations.

Other researchers have used the Gaussian sum approach in their work. Tam and Moore [36] developed a Gaussian sum estimator using extended Kalman filters in the problem of angle demodulation. Dmitriev and Shimelevich [37] applied the technique in determining the coordinates of a moving vehicle from geophysical field measurements. Sirisena and Brown [38] applied the algorithm to probabilistic and stochastic load flow problems. Namera and Stubberud [39] used the Gaussian sum approach in solving nonlinear fixed-point prediction problems. Gauvrit [40] developed a Gaussian sum filter to track targets in clutter with unknown noise variances. Tanaka and Katayama [41] devised a robust Kalman filter that consists of the Kalman filter plus MAP estimates of the selection parameters that specify noises from the Gaussian sum densities. Their filter is actually a smoother applied

to a linear system with Gaussian sum noises. Kitagawa [42] modeled sudden changes of trend or seasonal components due to the structural changes of the economic system by the presence of outliers by a nonGaussian model. He used Gaussian sum approximations of the nonGaussian densities.

2.2 Adaptive Kalman Filtering

In a typical application, the various parameters that describe the system are assumed to be known. Even if they are time-varying, the variation is assumed to be known. The parameters are then used in the Kalman filter design as the true model of the system. For many physical problems, the parameters may not be known exactly or may change at unknown times. In such cases, it is highly desirable to design the filter to be self-learning, so that it can adapt itself to the particular situation at hand.

One solution to this problem was formulated first by Magill [16]. He considered the problem of estimation of a Gaussian random process when some parameters of the process are initially unknown and remain constant with time. The parameters are assumed to come from a finite set of known values. The optimal adaptive estimate is a weighted sum of conditional estimates, which are formed by a bank of Kalman filters. Each Kalman filter is based on a particular parameter set. The weighting coefficients are determined by a nonlinear function of the measurement residuals of the filters. The measurement residual of the filter possessing the parameter set that matches the actual set will have the smallest expected value (typically, zero mean). The residuals of all the mismatched filters will be biased. Under the Gaussian assumption, the probability of the matched

filter will be the largest among all the filters.

Along similar lines, Ackerson and Fu [17] developed an adaptive state estimator for a linear system operating in a switching environment. The noise affecting the system comes from a group of Gaussian densities acting one at a time. The transitions from one noise source to the next is determined by a Markov transition probability matrix. This effort differed from that of Magill in that the parameter set describing the noise source does not remain constant with time, but is allowed to switch from one set to another at random intervals. The adaptive estimator has the same growing memory problem as that of Alspach, and consequently a suboptimal finite memory estimator was proposed.

Following Ackerson and Fu, Jaffer and Gupta [18] developed an adaptive estimator with a fixed number of filters for the problem of signal estimation under conditions of intermittent failure in the observations. Fujita and Fukao [19] established the validity of the separation theorem for the problem of determining an overall optimal control policy for a linear system with interrupted observations. Bruckner, Scott, and Rea [20] extended the adaptive algorithm of [17] to the switching of plant parameters in an air traffic control system.

A slightly different version of the adaptive filter was introduced by Moose [23]. He modeled the variations in the switching plant or switching environment by a semi-Markov process. Briefly stated, a semi-Markov process is a probabilistic system that makes its state transitions according to the transition probability matrix of a conventional Markov process. However, the amount of time spent in state i before the next transition to state j is a random variable [21], [22]. By incorporating the semi-Markov statistics into the learning portion of the adaptive filter, the problem of growing memory is completely eliminated.

This technique was successfully used by Moose and Wang [24] in the switching

plant problem. Other applications include: modeling large scale depth variations of submarines, Moose [25]; modeling accelerations of a maneuvering target in three-dimensional space using spherical observations of noisy radar data, Gholson and Moose [26]; incorporating a correlated acceleration model in an airborne target tracking system, Moose, VanLandingham, and McCabe [27]; passive underwater tracking using polar coordinates, McCabe and Moose [28]; using the correlated acceleration model in the passive underwater range tracking problem, Moose and Dailey [29]; extending the algorithm to passive underwater depth tracking, Moose and Godiwala [30]; modeling unknown biases in measurement devices, Moose, Sistanizadeh, and Skagfjord [31]; and incorporating a nonlinear system block in an underwater tracking system which decouples the bearing and range estimators, Moose [32].

Many others have contributed to the field of adaptive state estimation and multiple model filtering. Tugnait and Haddad [33], [34] studied state estimation in linear systems with random Markovian noise statistics. The noise comes from a group of Gaussian distributions with different means and covariances. The transitions are determined by an unknown Markov transition probability matrix. Akashi and Kumamoto [35] devised an adaptive filter in which the overall estimate is calculated using a relatively small number of individual conditional estimates sampled at random from a larger set of individual conditional estimates.

Maybeck and Suizu [46] and Tobin and Maybeck [47] used multiple model filtering in the problem of accurately tracking the azimuth and elevation of a highly maneuverable airborne target, using outputs from a forward-looking infrared sensor as measurements. Blom and Bar-Shalom [48] described a method of timing the merging of various individual filters in an adaptive estimator to overcome the growing memory problem. Bar-Shalom, Chang, and Blom [49] used the multiple

model filter in the problem of input estimation, providing estimates of the magnitude and onset time of the input. Emre and Seo [50] presented a global modeling approach to data association of multiple targets and maneuver detection/estimation of single targets. These two problems are solved simultaneously using system identification techniques, which leads to a multiple model estimator with a finite number of terms.

2.3 Alternate NonGaussian Filtering

Several alternatives to the Gaussian sum and adaptive Kalman filter algorithms have been proposed for systems containing nonGaussian inputs. Masreliez [43] introduced an approximate nonGaussian filtering method for linear systems. Instead of approximating the densities using Gaussian sums, he used a nonlinear score function of the measurement residuals to produce the Kalman filter equations. The technique works best however when only some of the random processes are nonGaussian, with the rest being Gaussian. Masreliez and Martin [44] extend this approximate nonGaussian filtering method by applying an influence function of min-max robustness theory to replace the previously used score function. Tsai and Kurz [45] devised a robust Kalman filter based on an m -interval polynomial approximation method for unknown nonGaussian noise. As the score function of [43] is partitioned into m segments, each is better approximated with a low-order polynomial function which is easier to implement and adapt to the density function of the residual process.

Other robust estimation techniques are used by Kirilin and Moghaddamjoo for systems with unknown inputs and nonGaussian measurement errors. In [51] the

input forcing function and measurement bias are estimated using a moving data window median calculation. The plant and measurement noise covariances are found using biweights and rank correlation. In [52] they discussed adaptive estimation of unknown inputs and measurement noise covariance using a running window curve-fitting algorithm. Estimation of the plant noise covariance is accomplished using an independent technique, based on the residuals and a stochastic approximation method.

3.0 NONGAUSSIAN ESTIMATION

This chapter discusses the mathematical details of the Gaussian sum approach to nonGaussian signal estimation. It begins with an outline of Alspach's development of the Gaussian sum estimation technique [6-8]. A major problem with this technique is indicated which necessitates a modified approach. Next, a detailed description of the nonGaussian signal model is given which leads to an alternate formulation of the Gaussian sum density approximation. The initial parameter vector describing the density approximation is found using a curve fitting procedure. A modified Gaussian sum adaptive filter is developed for a general discrete-time system and a simulation example is given. Finally, key differences between the Gaussian sum and modified Gaussian sum algorithms are examined.

3.1 Gaussian Sum Estimation Technique

As previously stated in chapter 2, an estimation technique, applicable to both linear systems with nonGaussian inputs and nonlinear systems with Gaussian

inputs, has been developed by Alspach [6-8]. The basic problem is to estimate the state x_k of a discrete-time system from the current measurement sequence $Z_k = \{z_1, z_2, \dots, z_{k-1}, z_k\}$. The "best" or optimal estimate \hat{x}_k of the system state x_k is chosen as that estimate which minimizes the mean-square error between x_k and \hat{x}_k . This results in the conditional mean estimate

$$\hat{x}_k = E[x_k | Z_k] = \int_{-\infty}^{\infty} x_k p(x_k | Z_k) dx_k \quad (3.1.1)$$

where $p(x_k | Z_k)$ is the conditional density function of the state given the current measurement sequence.

The Gaussian sum estimator is developed by Alspach using the following system model

$$x_k = \Phi_k x_{k-1} + w_{k-1} \quad (3.1.2)$$

$$z_k = H_k x_k + v_k \quad (3.1.3)$$

where the initial state has a Gaussian sum density of the form

$$p(x_0) = \sum_{i=1}^{\xi'_0} \alpha'_{0i} N[\mu'_{0i}, P'_{0i}] \quad (3.1.4)$$

Assume that the plant and measurement noise processes, w_k and v_k , are statistically independent, nonGaussian, white noise sequences with Gaussian sum densities

$$p(w_k) = \sum_{n=1}^{q_k} \beta_{kn} N[\omega_{kn}, Q_{kn}] \quad (3.1.5)$$

and

$$p(v_k) = \sum_{m=1}^{r_k} \gamma_{k_m} N[\nu_{k_m}, R_{k_m}] \quad (3.1.6)$$

The key idea of Gaussian sum estimation is to use (3.1.4) - (3.1.6) in the approximation of $p(x_k | Z_k)$ as a sum of Gaussian densities. Using Bayes' rule, this conditional density function can be determined recursively [7] from

$$p(x_k | Z_k) = \frac{p(x_k | Z_{k-1})p(z_k | x_k)}{p(z_k | Z_{k-1})} \quad (3.1.7)$$

$$p(x_k | Z_{k-1}) = \int_{-\infty}^{\infty} p(x_{k-1} | Z_{k-1})p(x_k | x_{k-1})dx_{k-1} \quad (3.1.8)$$

where the normalizing constant $p(z_k | Z_{k-1})$ in (3.1.7) is given by

$$p(z_k | Z_{k-1}) = \int_{-\infty}^{\infty} p(x_k | Z_{k-1})p(z_k | x_k)dx_k \quad (3.1.9)$$

The density $p(z_k | x_k)$ in (3.1.7) is determined by the measurement noise density $p(v_k)$ of (3.1.6) and the measurement equation (3.1.3). Similarly, the density $p(x_k | x_{k-1})$ in (3.1.8) is determined by the plant noise density $p(w_k)$ of (3.1.5) and the plant equation (3.1.2). Knowledge of $p(z_k | x_k)$, $p(x_k | x_{k-1})$, and the initial state density $p(x_0)$ of (3.1.4) determines $p(x_k | Z_k)$ for all k . Alspach [6], [7] derives two additional theorems which are briefly stated here. The measurement update theorem provides passage of $p(x_k | Z_{k-1})$ to $p(x_k | Z_k)$, while the time update theorem provides passage of $p(x_k | Z_k)$ to $p(x_{k+1} | Z_k)$.

Measurement Update Theorem

Suppose that the k^{th} stage prediction density function is

$$p(x_k | Z_{k-1}) = \sum_{i=1}^{\xi'_k} \alpha'_{k_i} N[\mu'_{k_i}, P'_{k_i}] \quad (3.1.10)$$

Then the k^{th} stage *filtering* density is given by

$$p(x_k | Z_k) = \sum_{j=1}^{\xi_k} \alpha_{k_j} N[\mu_{k_j}, P_{k_j}] \quad (3.1.11)$$

where the mean and covariance terms are found from the Kalman filter measurement update equations

$$\mu_{k_j} = \mu'_{k_i} + K_{k_j}(z_k - H_k \mu'_{k_i} - \nu_{k_m}) \quad (3.1.12)$$

$$P_{k_j} = [I - K_{k_j} H_k] P'_{k_i} \quad (3.1.13)$$

$$K_{k_j} = P'_{k_i} H_k^T (H_k P'_{k_i} H_k^T + R_{k_m})^{-1} \quad (3.1.14)$$

The filtering density (3.1.11) is the result of a double sum formed by the product of the prediction density (3.1.10) and the measurement noise density (3.1.6). The double sum is rewritten as a single sum with upper summation limit

$$\xi_k = \xi'_k r_k \quad (3.1.15)$$

produced by the product of upper summation limits ξ'_k of (3.1.10) and r_k of (3.1.6). As a result, the number of terms of the filtering Gaussian sum density grows geometrically at each measurement update with a ratio equal to the number of Gaussian sum terms of the measurement noise.

The α_{k_j} term of (3.1.11) is found from

$$\alpha_{k_j} = \frac{\alpha'_{k_i} \gamma_{k_m} W_{k_j}}{\sum_{i=1}^{\xi_k} \alpha'_{k_i} \gamma_{k_m} W_{k_j}} \quad (3.1.16)$$

where the index of summation j is defined as

$$j = i + (m - 1)\xi'_k \quad \begin{matrix} i = 1, \dots, \xi'_k \\ m = 1, \dots, r_k \end{matrix}$$

The W_{k_j} term of (3.1.16) is the Gaussian probability density function

$$W_{k_j} = N[H_k \mu'_{k_i} + \nu_{k_m}, H_k P'_{k_i} H_k^T + R_{k_m}] \quad (3.1.17)$$

with terms that are readily found from measurement update equations (3.1.12) and (3.1.14). The α_{k_j} term (3.1.16) is a nonlinear function of the current measurement data, with its denominator acting as a scale factor so that the α_{k_j} terms remain bounded between 0 and 1 at each iteration.

Time Update Theorem

Given that the k^{th} stage *filtering* density function is

$$p(x_k | Z_k) = \sum_{j=1}^{\xi_k} \alpha_{k_j} N[\mu_{k_j}, P_{k_j}] \quad (3.1.18)$$

Then the k^{th} stage *prediction* density is given by

$$p(x_{k+1} | Z_k) = \sum_{i=1}^{\xi'_{k+1}} \alpha'_{k+1,i} N[\mu'_{k+1,i}, P'_{k+1,i}] \quad (3.1.19)$$

where the mean and covariance terms are found from the Kalman filter time update equations

$$\mu'_{k+1,i} = \Phi_{k+1} \mu_{k_j} + \omega_{k_n} \quad (3.1.20)$$

$$P'_{k+1,i} = \Phi_{k+1} P_{k,j} \Phi_{k+1}^T + Q_{k,n} \quad (3.1.21)$$

The prediction density (3.1.19) is the result of a double sum formed by the product of the filtering density (3.1.18) and the plant noise density (3.1.5). The double sum is rewritten as a single sum with upper summation limit

$$\xi'_{k+1} = \xi_k q_k \quad (3.1.22)$$

produced by the product of upper summation limits ξ_k of (3.1.18) and q_k of (3.1.5). As a result, the number of terms of the prediction Gaussian sum density grows geometrically at each time update with a ratio equal to the number of Gaussian sum terms of the plant noise.

The $\alpha'_{k+1,i}$ term of (3.1.19) is found from

$$\alpha'_{k+1,i} = \alpha_{k,j} \beta_{k,n} \quad (3.1.23)$$

where the index of summation i is defined as

$$i = j + (n - 1)\xi_k \quad \begin{array}{l} j = 1, \dots, \xi_k \\ n = 1, \dots, q_k \end{array}$$

Alspach's two theorems show that for the system defined by (3.1.2) - (3.1.6) the Gaussian sum form repeats itself from one stage to the next so that (3.1.10) and (3.1.11) are the general forms for an arbitrary stage. The measurement update equations (3.1.12) - (3.1.14) and time update equations (3.1.20) and (3.1.21) comprise a bank of Kalman filters operating in parallel. Using the Gaussian sum approximation of the conditional density function (3.1.11), the overall state

estimate \hat{x}_k is found as a weighted sum of the individual conditional mean values from each filter

$$\hat{x}_k = E[x_k | Z_k] = \sum_{j=1}^{\xi_k} \alpha_{k_j} \mu_{k_j} \quad (3.1.24)$$

The corresponding conditional covariance is

$$P_k = E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T | Z_k] = \sum_{j=1}^{\xi_k} \alpha_{k_j} [P_{k_j} + (\hat{x}_k - \mu_{k_j})(\hat{x}_k - \mu_{k_j})^T] \quad (3.1.25)$$

It is evident from (3.1.15) and (3.1.22) that the number of Gaussian terms increases at each stage of the estimator. This *growing memory* problem is a serious limitation in Alspach's development. A modified approach is required to alleviate this limitation.

3.2 NonGaussian Signal Model Development

Let u be a random noise process or random input signal with a nonGaussian density function. It can be modeled as the sum of two statistically independent random processes

$$u = b + n \quad (3.2.1)$$

The first term, b , is a semi-Markov process with state transitions governed by the transition probability matrix of a conventional Markov process. Markov processes have the property that a transition is made at every time instant. The transition

may return the process to the state it previously occupied, but a transition occurs nevertheless. However, in the semi-Markov case, the amount of time between transitions is a random variable [22]. The value of b is randomly selected from a fixed set of discrete values, characterized by a delta probability density function

$$p(b) = \sum_{i=1}^M P_i \delta(b - b_i) \quad (3.2.2)$$

with

$$\sum_{i=1}^M P_i = 1; \quad P_i \geq 0 \text{ for } i = 1, 2, \dots, M \quad (3.2.3)$$

This process can be thought of as a randomly-switching bias, each bias value b_i having probability P_i .

The second term, n , is a zero mean white Gaussian process with variance σ_n^2 . With both densities known, the density function of u can be found using the convolution relationship between u , n , and b [53]

$$p(u) = \int_{-\infty}^{\infty} p_n(u - b) p_b(b) db \quad (3.2.4)$$

where $p_n(u - b)$ is the Gaussian density with $n = u - b$

$$p_n(u - b) = \frac{1}{\sqrt{2\pi}\sigma_n} e^{-\frac{1}{2}\left(\frac{u-b}{\sigma_n}\right)^2} \quad (3.2.5)$$

Substituting (3.2.2) and (3.2.5) into the convolution integral (3.2.4) gives

$$p(u) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_n} e^{-\frac{1}{2}\left(\frac{u-b}{\sigma_n}\right)^2} \sum_{i=1}^M P_i \delta(b - b_i) db \quad (3.2.6)$$

Interchanging integration and summation

$$p(u) = \sum_{i=1}^M P_i \left[\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_n} e^{-\frac{1}{2}\left(\frac{u-b}{\sigma_n}\right)^2} \delta(b-b_i) db \right] \quad (3.2.7)$$

Using the sifting property of the delta function [54]

$$f(a) = \int_{-\infty}^{\infty} f(x) \delta(x-a) dx \quad (3.2.8)$$

the integral of (3.2.7) is evaluated as

$$p(u) = \sum_{i=1}^M P_i \left[\frac{1}{\sqrt{2\pi}\sigma_n} e^{-\frac{1}{2}\left(\frac{u-b_i}{\sigma_n}\right)^2} \right] \quad (3.2.9)$$

or

$$p(u) = \sum_{i=1}^M P_i N[b_i, \sigma_n^2] \quad (3.2.10)$$

Thus, the nonGaussian density function of u can be modeled as a Gaussian sum. The weight P_i of each Gaussian term is the probability of the i^{th} bias term. The bias term b_i is restricted to be randomly selected from the same fixed set of bias values at each iteration. Using this model in conjunction with the modified Gaussian sum adaptive filter developed in the next section avoids the growing memory problem of Alspach's development.

Selecting the parameters P_i , b_i , and σ_n in (3.2.10) to obtain the "best" approximation p_{GS} to some actual nonGaussian density function p_A is accomplished by means of minimizing the L^k norm

$$\|p_A - p_{GS}\|^k = \int_{-\infty}^{\infty} \left| p_A(u) - \sum_{i=1}^M P_i N[b_i, \sigma_n^2] \right|^k du \quad (3.2.11)$$

This curve fitting exercise can be done off-line using several values of M until a suitable trade-off between minimum norm and minimum M is obtained. Alspach [6], [7] performed this curve fitting procedure using L^1 and L^2 norms for a uniform density and a Gamma density. It was found that minimizing the L^2 norm resulted in many fewer terms in the Gaussian sum and a considerably better looking approximation for both densities compared to minimizing the L^1 norm.

Figure 1 compares a Gamma density with a four-term Gaussian sum density approximation minimizing the L^2 norm. The Gaussian sum curve is shown to fit the Gamma curve reasonably well. The Gamma density used is

$$p(u) = \begin{cases} \frac{u^3 e^{-u}}{6} & u \geq 0 \\ 0 & u < 0 \end{cases} \quad (3.2.12)$$

Each term of the Gaussian sum has a fixed value of $\sigma_n = 1$. Table 1 lists the values of P_i and b_i used in the Gaussian sum.

Table 1.
Gaussian sum P_i , b_i values, $\sigma_n = 1$.

| i | P_i | b_i |
|-----|-------|-------|
| 1 | 0.081 | 2.537 |
| 2 | 0.432 | 2.553 |
| 3 | 0.356 | 4.555 |
| 4 | 0.131 | 6.933 |

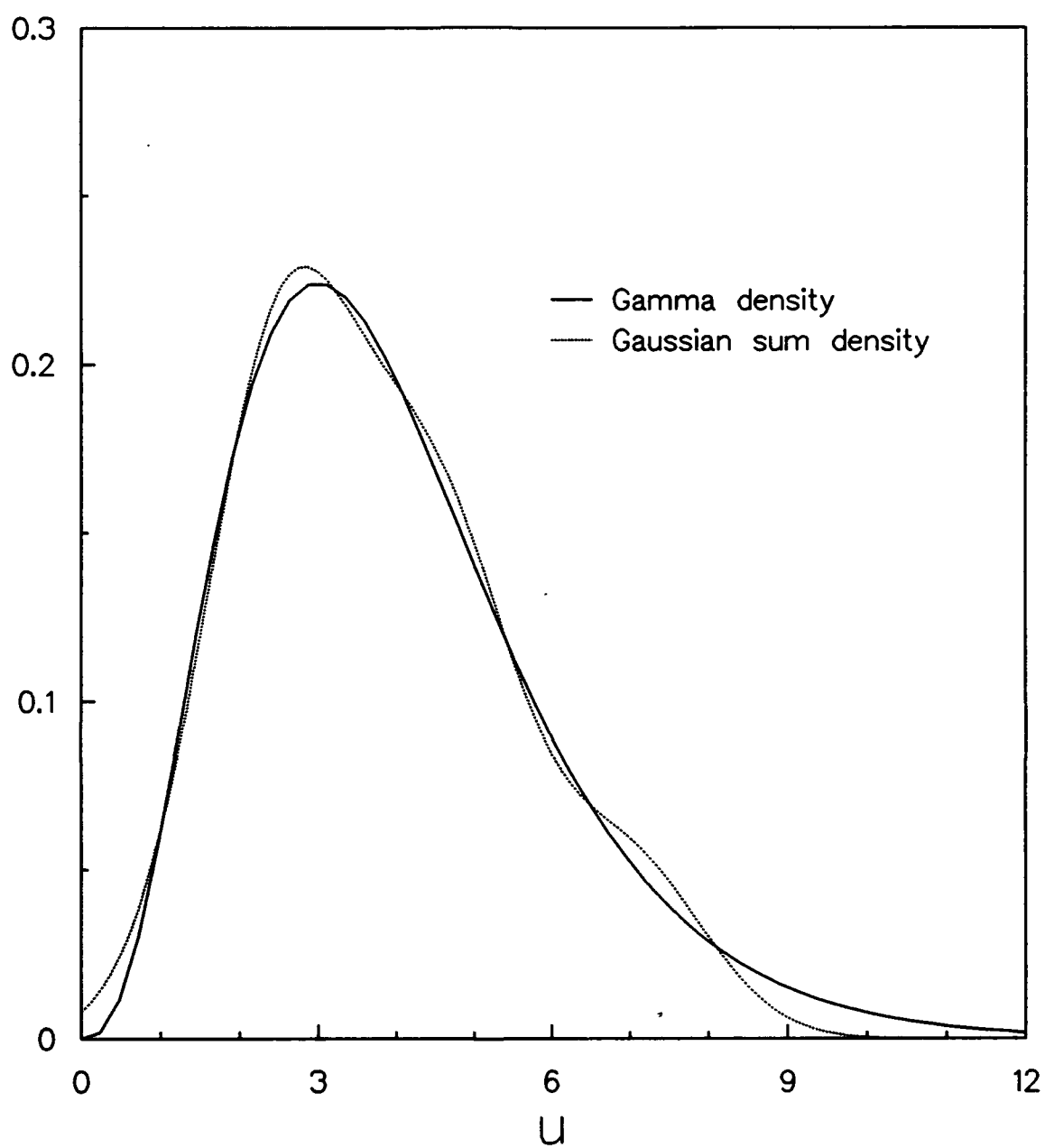


Figure 1. Gamma and Gaussian sum probability density functions.

3.3 Modified Gaussian Sum Adaptive Filtering

A modified Gaussian sum adaptive filter is now developed for a linear system with a deterministic input signal and nonGaussian plant and measurement noise. The system is modeled in standard discrete-time state-space form as

$$x_{k+1} = \Phi x_k + \Gamma u_k + \Psi w_k \quad (3.3.1)$$

$$w_k = a_k + m_k \quad (3.3.2)$$

$$z_k = H x_k + v_k \quad (3.3.3)$$

$$v_k = b_k + n_k \quad (3.3.4)$$

where

x_{k+1} is the state vector

u_k is a known deterministic input

w_k is the vector Gaussian sum signal model of the actual nonGaussian plant noise process, comprised of semi-Markov bias vector a_k , and zero mean white Gaussian noise m_k with covariance Q

z_k is the measurement vector

v_k is the vector Gaussian sum signal model of the actual nonGaussian measurement noise process, comprised of semi-Markov bias vector b_k , and zero mean white Gaussian noise n_k with covariance R

Φ, Γ, Ψ, H are the respective constant transition matrices

and the random quantities x_k, a_k, m_k, b_k, n_k are assumed to be mutually statistically independent.

The optimal estimate \hat{x}_{k+1} of the state vector is found by minimizing the mean-square error between x_{k+1} and \hat{x}_{k+1} . This results in the conditional mean estimate

$$\hat{x}_{k+1} = E[x_{k+1} | Z_{k+1}] = \int_{-\infty}^{\infty} x_{k+1} p(x_{k+1} | Z_{k+1}) dx_{k+1} \quad (3.3.5)$$

where Z_{k+1} is the current measurement sequence $\{z_1, z_2, \dots, z_{k+1}\}$. The conditional density function of (3.3.5) can be written as the ratio of the corresponding joint and marginal densities

$$p(x_{k+1} | Z_{k+1}) = \frac{p(x_{k+1}, Z_{k+1})}{p(Z_{k+1})} \quad (3.3.6)$$

The two bias vectors a and b are explicitly brought into (3.3.6) by considering the joint density $p(x_{k+1}, Z_{k+1})$ to be a marginal density found from $p(x_{k+1}, Z_{k+1}, a = a_i, b = b_j)$ by summing over the a and b terms

$$p(x_{k+1} | Z_{k+1}) = \frac{\sum_{i=1}^N \sum_{j=1}^M p(x_{k+1}, Z_{k+1}, a = a_i, b = b_j)}{p(Z_{k+1})} \quad (3.3.7)$$

Expanding (3.3.7) and using $p(x_{k+1}, Z_{k+1}, a_i, b_j)$ as shorthand for $p(x_{k+1}, Z_{k+1}, a = a_i, b = b_j)$ gives

$$p(x_{k+1} | Z_{k+1}) = \frac{\sum_{i=1}^N \sum_{j=1}^M p(x_{k+1} | Z_{k+1}, a_i, b_j) p(Z_{k+1}, a_i, b_j)}{p(Z_{k+1})}$$

$$p(x_{k+1} | Z_{k+1}) = \frac{\sum_{i=1}^N \sum_{j=1}^M p(x_{k+1} | Z_{k+1}, a_i, b_j) p(a_i, b_j | Z_{k+1}) p(Z_{k+1})}{p(Z_{k+1})}$$

and finally

$$p(x_{k+1} | Z_{k+1}) = \sum_{i=1}^N \sum_{j=1}^M p(x_{k+1} | Z_{k+1}, a_i, b_j) p(a_i, b_j | Z_{k+1}) \quad (3.3.8)$$

Substituting (3.3.8) into the conditional mean (3.3.5) results in

$$\hat{x}_{k+1} = \int_{-\infty}^{\infty} x_{k+1} \left[\sum_{i=1}^N \sum_{j=1}^M p(x_{k+1} | Z_{k+1}, a_i, b_j) p(a_i, b_j | Z_{k+1}) \right] dx_{k+1} \quad (3.3.9)$$

Interchanging integration and summation gives

$$\hat{x}_{k+1} = \sum_{i=1}^N \sum_{j=1}^M \left[\int_{-\infty}^{\infty} x_{k+1} p(x_{k+1} | Z_{k+1}, a_i, b_j) dx_{k+1} \right] p(a_i, b_j | Z_{k+1}) \quad (3.3.10)$$

The bracketed integral in (3.3.10) is the conditional mean estimate of x_{k+1} given that $a = a_i$ and $b = b_j$, denoted by

$$\hat{x}_{k+1}^{ij} = \int_{-\infty}^{\infty} x_{k+1} p(x_{k+1} | Z_{k+1}, a_i, b_j) dx_{k+1} \quad (3.3.11)$$

In effect, \hat{x}_{k+1}^{ij} represents the estimate for the ij^{th} density combination from the two Gaussian sums. A set of $N \times M$ (NM) estimators is needed to provide all the individual \hat{x}_{k+1}^{ij} estimates. The overall estimate from (3.3.10) and (3.3.11),

$$\hat{x}_{k+1} = \sum_{i=1}^N \sum_{j=1}^M \hat{x}_{k+1}^{ij} p(a_i, b_j | Z_{k+1}) \quad (3.3.12)$$

is a weighted sum of the NM individual estimates. The weighting factor $p(a_i, b_j | Z_{k+1})$ is the probability that $a = a_i$ and $b = b_j$ given the current measurement sequence.

Each estimate (3.3.11) is found from a modified linear state estimator. Rewriting the measurement sequence as $Z_{k+1} = \{Z_k, z_{k+1}\}$, the conditional density function of (3.3.11) is

$$p(x_{k+1} | Z_k, z_{k+1}, a_i, b_j) = \frac{p(x_{k+1}, Z_k, z_{k+1}, a_i, b_j)}{p(Z_k, z_{k+1}, a_i, b_j)} \quad (3.3.13)$$

Expanding as before gives

$$p(x_{k+1} | Z_k, z_{k+1}, a_i, b_j) = \frac{p(z_{k+1} | x_{k+1}, a_i, b_j, Z_k) p(x_{k+1} | a_i, b_j, Z_k)}{p(z_{k+1} | a_i, b_j, Z_k)} \quad (3.3.14)$$

The first term of the numerator of (3.3.14) is Gaussian since x_{k+1} , $a = a_i$, and $b = b_j$ are given. The conditional mean value is found by first combining (3.3.3) and (3.3.4) at time iteration $k+1$

$$z_{k+1} = Hx_{k+1} + b_{k+1} + n_{k+1} \quad (3.3.15)$$

and then taking the expected value

$$m_{k+1}^{ij} = E[z_{k+1} | x_{k+1}, a_i, b_j, Z_k] = Hx_{k+1} + b_j + E[n_{k+1} | x_{k+1}, a_i, b_j, Z_k]$$

Since n_{k+1} has a mean value of zero, the expected value is

$$m_{k+1}^{ij} = Hx_{k+1} + b_j \quad (3.3.16)$$

The conditional covariance is found next as

$$\begin{aligned} C_{k+1}^{ij} &= E[(z_{k+1} - m_{k+1}^{ij})(z_{k+1} - m_{k+1}^{ij})^T | x_{k+1}, a_i, b_j, Z_k] \\ &= E[(Hx_{k+1} + b_{k+1} + n_{k+1} - Hx_{k+1} - b_j)(z_{k+1} - m_{k+1}^{ij})^T | x_{k+1}, a_i, b_j, Z_k] \end{aligned}$$

Knowing that $b = b_j$

$$C_{k+1}^{ij} = E[n_{k+1}n_{k+1}^T | x_{k+1}, a_i, b_j, Z_k] = R \quad (3.3.17)$$

The conditional density is therefore

$$p(z_{k+1} | x_{k+1}, a_i, b_j, Z_k) = N[m_{k+1}^{ij}, R] = N[Hx_{k+1} + b_j, R] \quad (3.3.18)$$

The second term of the numerator of (3.3.14) is also Gaussian since $a = a_i$ and $b = b_j$ are given. The conditional mean value is found by first combining (3.3.1) and (3.3.2)

$$x_{k+1} = \Phi x_k + \Gamma u_k + \Psi a_k + \Psi m_k \quad (3.3.19)$$

and then taking the expected value

$$\tilde{x}_{k+1}^{ij} = E[x_{k+1} | a_i, b_j, Z_k] = \Phi E[x_k | a_i, b_j, Z_k] + \Gamma u_k + \Psi a_i + \Psi E[m_k | a_i, b_j, Z_k]$$

Since m_k has a mean value of zero, the expected value is

$$\tilde{x}_{k+1}^{ij} = \Phi \hat{x}_k^{ij} + \Gamma u_k + \Psi a_i \quad (3.3.20)$$

where \tilde{x}_{k+1}^{ij} is the *prediction* at time iteration $k+1$ given prior measurements up to only iteration k , while \hat{x}_k^{ij} is the previous *estimate* at iteration k given measurements up to iteration k .

The conditional covariance is found next as

$$\begin{aligned} M_{k+1}^{ij} &= E[(x_{k+1} - \tilde{x}_{k+1}^{ij})(x_{k+1} - \tilde{x}_{k+1}^{ij})^T | a_i, b_j, Z_k] \\ &= E[(\Phi x_k + \Gamma u_k + \Psi a_k + \Psi m_k - \Phi \hat{x}_k^{ij} - \Gamma u_k - \Psi a_i)(x_{k+1} - \tilde{x}_{k+1}^{ij})^T | a_i, b_j, Z_k] \end{aligned}$$

Since $a = a_i$ and recalling that x_k and m_k are statistically independent

$$M_{k+1}^{ij} = \Phi E[(x_k - \hat{x}_k^{ij})(x_k - \hat{x}_k^{ij})^T | a_i, b_j, Z_k] \Phi^T + \Psi E[m_k m_k^T | a_i, b_j, Z_k] \Psi^T$$

Letting $P_k^{ij} = E[(x_k - \hat{x}_k^{ij})(x_k - \hat{x}_k^{ij})^T | a_i, b_j, Z_k]$ be the conditional covariance of the error term $x_k - \hat{x}_k^{ij}$,

$$M_{k+1}^{ij} = \Phi P_k^{ij} \Phi^T + \Psi Q \Psi^T \quad (3.3.21)$$

Since the covariances Q and R are the same for each ij^{th} estimator, the respective conditional covariances simplify to $P_k^{ij} = P_k$ and $M_{k+1}^{ij} = M_{k+1}$, and the conditional density is therefore

$$p(x_{k+1} | a_i, b_j, Z_k) = N[\tilde{x}_{k+1}^{ij}, M_{k+1}] = N[\Phi \hat{x}_k^{ij} + \Gamma u_k + \Psi a_i, \Phi P_k \Phi^T + \Psi Q \Psi^T] \quad (3.3.22)$$

The least-mean-square (LMS) estimator is found by taking the derivative of the natural logarithm of (3.3.14) with respect to x_{k+1} , setting the result equal to zero, and determining the value of $x_{k+1} = \hat{x}_{k+1}^{ij}$ that produces this result. Since the

denominator term of (3.3.14) does not depend on x_{k+1} , it will not play a role in determining \hat{x}_{k+1}^{ij} . Now, set

$$\frac{\partial}{\partial x_{k+1}} [\ln p(x_{k+1} | Z_k, z_{k+1}, a_i, b_j)] = 0 \quad (3.3.23)$$

Using (3.3.18) and (3.3.22) in (3.3.14), (3.3.23) becomes

$$\frac{\partial}{\partial x_{k+1}} [\ln N[m_{k+1}^{ij}, R] + \ln N[\tilde{x}_{k+1}^{ij}, M_{k+1}] - 0] = 0$$

$$\frac{\partial}{\partial x_{k+1}} \left[-\frac{1}{2}(z_{k+1} - m_{k+1}^{ij})^T R^{-1}(z_{k+1} - m_{k+1}^{ij}) - \frac{1}{2}(x_{k+1} - \tilde{x}_{k+1}^{ij})^T M_{k+1}^{-1}(x_{k+1} - \tilde{x}_{k+1}^{ij}) \right] = 0 \quad (3.3.24)$$

Combining (3.3.16) and using a rule of matrix differentiation (if $C = C^T$)

$$\frac{\partial}{\partial x} (Ax + b)^T C (Ax + b) = 2A^T C (Ax + b) \quad (3.3.25)$$

the first term of (3.3.24) is arranged in the form of (3.3.25)

$$\frac{\partial}{\partial x_{k+1}} \left[-\frac{1}{2}(-Hx_{k+1} + z_{k+1} - b_j)^T R^{-1}(-Hx_{k+1} + z_{k+1} - b_j) \right]$$

with $A = -H$, $b = z_{k+1} - b_j$, and $C = R^{-1}$. Applying (3.3.25) results in

$$-H^T R^{-1}(Hx_{k+1} - z_{k+1} + b_j) \quad (3.3.26)$$

Similarly, the second term of the derivative (3.2.24) is

$$-M_{k+1}^{-1}(x_{k+1} - \tilde{x}_{k+1}^{ij}) \quad (3.3.27)$$

Combining (3.3.20), (3.3.26), and (3.3.27) and setting $x_{k+1} = \hat{x}_{k+1}^{ij}$ gives

$$-H^T R^{-1}(H \hat{x}_{k+1}^{ij} - z_{k+1} + b_j) - M_{k+1}^{-1}(\hat{x}_{k+1}^{ij} - \Phi \hat{x}_k^{ij} - \Gamma u_k - \Psi a_i) = 0$$

or

$$(M_{k+1}^{-1} + H^T R^{-1} H) \hat{x}_{k+1}^{ij} = H^T R^{-1}(z_{k+1} - b_j) + M_{k+1}^{-1}(\Phi \hat{x}_k^{ij} + \Gamma u_k + \Psi a_i) \quad (3.3.28)$$

Now, define

$$P_{k+1}^{-1} = (M_{k+1}^{-1} + H^T R^{-1} H) \quad (3.3.29)$$

Using (3.3.29) and solving for \hat{x}_{k+1}^{ij} in (3.3.28) gives

$$\hat{x}_{k+1}^{ij} = P_{k+1} H^T R^{-1}(z_{k+1} - b_j) + P_{k+1} M_{k+1}^{-1}(\Phi \hat{x}_k^{ij} + \Gamma u_k + \Psi a_i) \quad (3.3.30)$$

To rewrite $P_{k+1} M_{k+1}^{-1}$, first premultiply (3.3.29) by P_{k+1} , giving the identity matrix

$$I = P_{k+1}(M_{k+1}^{-1} + (H^T R^{-1} H)) \quad (3.3.31)$$

then solve (3.3.31) for $P_{k+1} M_{k+1}^{-1}$ and substitute into (3.3.30)

$$\hat{x}_{k+1}^{ij} = P_{k+1} H^T R^{-1}(z_{k+1} - b_j) + (I - P_{k+1} H^T R^{-1} H)(\Phi \hat{x}_k^{ij} + \Gamma u_k + \Psi a_i) \quad (3.3.32)$$

Now, let

$$K_{k+1} = P_{k+1} H^T R^{-1} \quad (3.3.33)$$

be the Kalman gain, and rearrange using (3.3.29) to produce

$$\begin{aligned} K_{k+1} &= (M_{k+1} + H^{-1}RH^T)^{-1}H^TR^{-1} \\ &= M_{k+1}H^TR^{-1} + H^{-1} \\ &= (M_{k+1}H^T(R^{-1} + H^T^{-1}M_{k+1}^{-1}H^{-1}))^{-1} \end{aligned}$$

or

$$K_{k+1} = M_{k+1}H^T(HM_{k+1}H^T + R)^{-1} \quad (3.3.34)$$

Substituting (3.3.33) into (3.3.32) and rearranging produces the familiar Kalman filter equation below, with modifications to allow for the two bias terms of the Gaussian sum densities

$$\hat{x}_{k+1}^{ij} = \Phi \hat{x}_k^{ij} + \Gamma u_k + \Psi a_i + K_{k+1}[z_{k+1} - b_j - H(\Phi \hat{x}_k^{ij} + \Gamma u_k + \Psi a_i)] \quad (3.3.35)$$

As a final step, substitute (3.3.33) into (3.3.31) to produce

$$P_{k+1}M_{k+1}^{-1} = (I - K_{k+1}H)$$

or

$$P_{k+1} = (I - K_{k+1}H)M_{k+1} \quad (3.3.36)$$

With the individual ij^{th} estimate (3.3.35) determined, the next step in producing the overall estimate (3.3.12) is to find the weighting term $p(a_i, b_j | Z_{k+1})$. Using $Z_{k+1} = \{Z_k, z_{k+1}\}$ again, the weighting term becomes

$$p(a_i, b_j | Z_{k+1}) = \frac{p(z_{k+1}, a_i, b_j, Z_k)}{p(z_{k+1}, Z_k)}$$

Using Bayes' theorem produces

$$p(a_i, b_j | Z_{k+1}) = \frac{p(z_{k+1} | a_i, b_j, Z_k) p(a_i, b_j | Z_k)}{p(z_{k+1} | Z_k)} \quad (3.3.37)$$

The first term of the numerator can be modeled as a Gaussian density if the bias terms switch slowly compared to the time interval k . This assumption will be made here and has been validated by extensive simulation and analysis by Moose and Wang [24]. Combining (3.3.15) and (3.3.19), z_{k+1} is written as

$$z_{k+1} = H(\Phi x_k + \Gamma u_k + \Psi a_k + \Psi m_k) + b_{k+1} + n_{k+1} \quad (3.3.38)$$

Taking the expected value, the conditional mean is

$$\begin{aligned} \tilde{z}_{k+1}^{ij} &= E[z_{k+1} | a_i, b_j, Z_k] \\ &= H\Phi E[x_k | a_i, b_j, Z_k] + H\Gamma u_k + H\Psi a_i + b_j \end{aligned}$$

or

$$\tilde{z}_{k+1}^{ij} = H(\Phi \hat{x}_k^{ij} + \Gamma u_k + \Psi a_i) + b_j \quad (3.3.39)$$

The conditional covariance is found next as

$$R_{k+1}^{ij} = E[(z_{k+1} - \tilde{z}_{k+1}^{ij})(z_{k+1} - \tilde{z}_{k+1}^{ij})^T | a_i, b_j, Z_k] \quad (3.3.40)$$

Using (3.3.38) and (3.3.39), the $z_{k+1} - \tilde{z}_{k+1}^{ij}$ term is written as

$$z_{k+1} - \tilde{z}_{k+1}^{ij} = H(\Phi x_k + \Gamma u_k + \Psi a_k + \Psi m_k) + b_{k+1} + n_{k+1} - H(\Phi \hat{x}_k^{ij} + \Gamma u_k + \Psi a_i) - b_j \quad (3.3.41)$$

Inserting (3.3.41) into (3.3.40) gives

$$R_{k+1}^{ij} = E[(H\{\Phi(x_k - \hat{x}_k^{ij}) + \Psi m_k\} + n_{k+1})(z_{k+1} - \tilde{z}_{k+1}^{ij})^T | a_i, b_j, Z_k]$$

and using (3.3.17) and (3.3.21) produces

$$R_{k+1}^{ij} = HM_{k+1}^{ij}H^T + R \quad (3.3.42)$$

Since the covariances R and $M_{k+1}^{ij} = M_{k+1}$ are the same for each ij^{th} estimator, the conditional covariance simplifies to $R_{k+1}^{ij} = R_{k+1}$, and the conditional density is therefore

$$\begin{aligned} p(z_{k+1} | a_i, b_j, Z_k) &= N[\tilde{z}_{k+1}^{ij}, R_{k+1}] \\ &= N[H(\Phi \hat{x}_k^{ij} + \Gamma u_k + \Psi a_i) + b_j, HM_{k+1}H^T + R] \end{aligned} \quad (3.3.43)$$

The second term of the numerator of (3.3.37) can be rewritten to explicitly include the time interval for the two bias terms

$$p(a_i, b_j | Z_k) = p(a_{k+1} = a_i, b_{k+1} = b_j | Z_k) \quad (3.3.44)$$

Using Bayes' theorem and the definition of marginal densities, (3.3.44) is expanded into

$$\begin{aligned} p(a_i, b_j | Z_k) &= \frac{\sum_{\alpha=1}^N \sum_{\beta=1}^M p(a_{k+1} = a_i, a_k = a_\alpha, b_{k+1} = b_j, b_k = b_\beta, Z_k)}{p(Z_k)} \\ &= \sum_{\alpha=1}^N \sum_{\beta=1}^M p(a_{k+1} = a_i, b_{k+1} = b_j | a_k = a_\alpha, b_k = b_\beta, Z_k) p(a_k = a_\alpha, b_k = b_\beta, Z_k) \end{aligned} \quad (3.3.45)$$

Recalling that a_{k+1} and b_{k+1} are assumed statistically independent, and given that Z_k , $a_k = a_\alpha$, and $b_k = b_\beta$ are known, (3.3.45) becomes

$$= \sum_{\alpha=1}^N \sum_{\beta=1}^M p(a_{k+1} = a_i | a_k = a_\alpha) p(b_{k+1} = b_j | b_k = b_\beta) p(a_k = a_\alpha, b_k = b_\beta | Z_k) \quad (3.3.46)$$

The measurement sequence Z_k is not needed in the first two terms of (3.3.46) since $a_k = a_\alpha$ and $b_k = b_\beta$ are known at time interval k and are independent of Z_k . The following notation will be used to express the three terms of the summation (3.3.46)

$$\theta_a^{i\alpha} = p(a_{k+1} = a_i | a_k = a_\alpha) \quad (3.3.47)$$

$$\theta_b^{j\beta} = p(b_{k+1} = b_j | b_k = b_\beta) \quad (3.3.48)$$

$$w_k^{\alpha\beta} = p(a_k = a_\alpha, b_k = b_\beta | Z_k) \quad (3.3.49)$$

The terms (3.3.47) and (3.3.48) are Markov transition probabilities [22]; that is, $\theta_a^{i\alpha}$ is the conditional probability that $a = a_i$ at time interval $k+1$, given that $a = a_\alpha$ at time interval k . The $\theta_b^{j\beta}$ term is similarly defined. The term (3.3.49) is of the same form as (3.3.37) and is just the previous weighting term at the previous time interval k .

The denominator term of (3.3.37) is independent of ij . Therefore it is the same for each ij^{th} estimator and becomes a scale factor.

Combining (3.3.43) and (3.3.47) - (3.3.49), the weighting term (3.3.37) is written as

$$w_{k+1}^{ij} = p(a_i, b_j | Z_{k+1}) = C_{k+1}^o N[\tilde{z}_{k+1}^{ij}, R_{k+1}] \sum_{\alpha=1}^N \sum_{\beta=1}^M \theta_a^{i\alpha} \theta_b^{j\beta} w_k^{\alpha\beta} \quad (3.3.50)$$

where C_{k+1}^o is a scale factor determined at each time interval such that

$$\sum_{i=1}^N \sum_{j=1}^M w_{k+1}^{ij} = 1 \quad (3.3.51)$$

guaranteeing that the sum of all the weighting terms (3.3.50) is equal to one. The mean and covariance of the Gaussian density function are available from the Kalman filter equation (3.3.35) and Kalman gain (3.3.34).

The equations for the modified Gaussian sum adaptive filter for the system of (3.3.1) - (3.3.4) are summarized in Table 2. The structure of the overall adaptive filter is a bank of Kalman filters operating in parallel, with each individual estimate multiplied by its own corresponding weighting term. The ij^{th} estimator based on the bias terms that most closely matches the actual bias terms of the modeled system will have a corresponding weighting term that tends closer to one, while the weights of the other mismatched estimators will tend towards zero. A block diagram of the adaptive filter is shown in Figure 2.

A simulation example is presented in the following section.

Table 2.
Modified Gaussian sum adaptive filter equations.

System:

$$\begin{aligned}x_{k+1} &= \Phi x_k + \Gamma u_k + \Psi w_k \\w_k &= a_k + m_k \\z_k &= H x_k + v_k \\v_k &= b_k + n_k\end{aligned}$$

Overall estimate:

$$\hat{x}_{k+1} = \sum_{i=1}^N \sum_{j=1}^M \hat{x}_{k+1}^{ij} p(a_i, b_j | Z_{k+1})$$

Kalman filter equation:

$$\hat{x}_{k+1}^{ij} = \Phi \hat{x}_k^{ij} + \Gamma u_k + \Psi a_i + K_{k+1} [z_{k+1} - b_j - H(\Phi \hat{x}_k^{ij} + \Gamma u_k + \Psi a_i)]$$

Kalman gain equations: $M_{k+1} = \Phi P_k \Phi^T + \Psi Q \Psi^T$

$$K_{k+1} = M_{k+1} H^T (H M_{k+1} H^T + R)^{-1}$$

$$P_{k+1} = (I - K_{k+1} H) M_{k+1}$$

Weighting term:

$$w_{k+1}^{ij} = p(a_i, b_j | Z_{k+1}) = C_{k+1}^o p(z_{k+1} | a_i, b_j, Z_k) \sum_{\alpha=1}^N \sum_{\beta=1}^M \theta_a^{i\alpha} \theta_b^{j\beta} w_k^{\alpha\beta}$$

with $p(z_{k+1} | a_i, b_j, Z_k) = N[H(\Phi \hat{x}_k^{ij} + \Gamma u_k + \Psi a_i) + b_j, H M_{k+1} H^T + R]$

$$\theta_a^{i\alpha} = p(a_{k+1} = a_i | a_k = a_\alpha)$$

$$\theta_b^{j\beta} = p(b_{k+1} = b_j | b_k = b_\beta)$$

$$w_k^{\alpha\beta} = p(a_k = a_\alpha, b_k = b_\beta | Z_k)$$

and scale factor C_{k+1}^o such that $\sum_{i=1}^N \sum_{j=1}^M w_{k+1}^{ij} = 1$

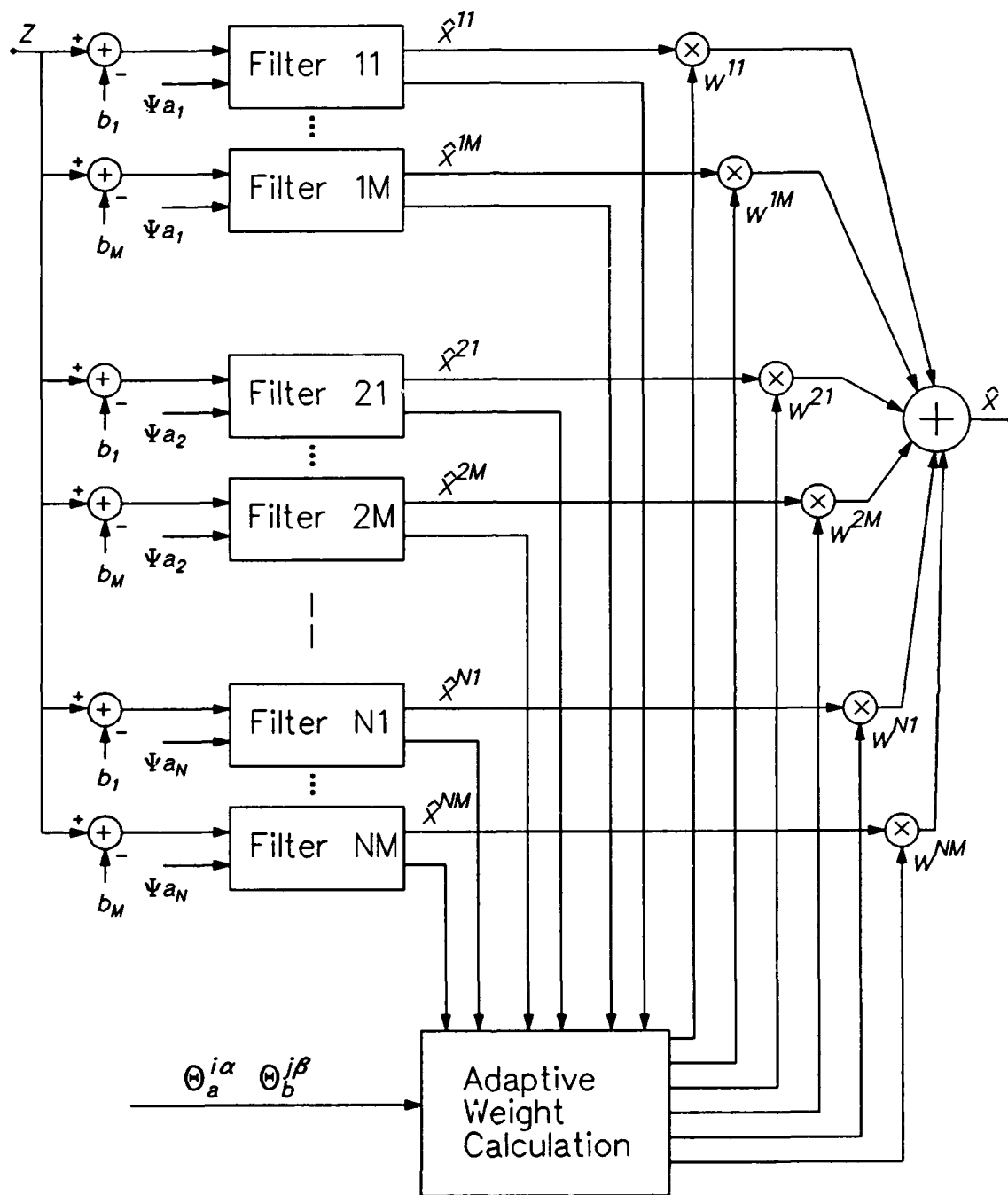


Figure 2. Modified Gaussian sum adaptive filter structure.

3.4 Simulation Example

An example illustrating the modified Gaussian sum estimation technique is now presented. A first-order system is used, modeled by the following discrete-time equations

$$\text{plant:} \quad x_{k+1} = e^{-\alpha T} x_k + (1 - e^{-\alpha T}) u_k + \frac{(1 - e^{-\alpha T})}{\alpha} w_k \quad (3.4.1)$$

$$w_k = a_k + m_k \quad (3.4.2)$$

$$\text{measurement:} \quad z_k = x_k + v_k \quad (3.4.3)$$

$$v_k = b_k + n_k \quad (3.4.4)$$

The value of α is 0.6 and the sample time $T = 1$ second. For simplicity, let the plant noise (3.4.2) be zero mean white Gaussian with variance $Q = 1.0$. Therefore, the randomly-switching plant bias term $a_k = 0$. The actual measurement noise, modeled by (3.4.4), has the Gamma density of (3.2.12), with a mean and variance of 4. The measurement bias term b_k can be randomly selected from the last three bias terms of Table 1, {2.553, 4.555, 6.933}. The first bias term of Table 1, {2.537}, is not used since it is so close in value to the second bias term. The measurement model noise term n_k is zero mean white Gaussian with variance $R = 1.0$. A deterministic input of $u_k = 10$ is used throughout the simulation. Figure 3a shows the measurement and state sequences. Note how the measurement is centered about 14, indicating a mean value for the Gamma density of 4.

The filter is initialized with equally-valued weighting terms

$$w_0^j = \frac{1}{3}, \quad \text{for } j = 1, 2, 3 \quad (3.4.5)$$

A Markov transition probability matrix $\Theta_b^{j\beta}$, consisting of $\theta_b^{j\beta}$ elements, is configured with a high probability that the bias term does not switch from one value to another, and a low probability that the bias term does switch, such as

$$\Theta_b^{j\beta} = \begin{bmatrix} .95 & .025 & .025 \\ .025 & .95 & .025 \\ .025 & .025 & .95 \end{bmatrix} \quad (3.4.6)$$

The initial value of the state and state estimate is $x_0 = \hat{x}_0 = 20$, and the initial value of the variance of the error $x_{k+1} - \hat{x}_{k+1}$ is $P_o = 100$. The overall state estimate and the state are shown in Figure 3b, with the error and overall state estimate shown in Figure 3c. Note how the error appears to be zero mean, thus showing that the MGS adaptive filter removes the bias effect of the nonzero mean Gamma measurement noise.

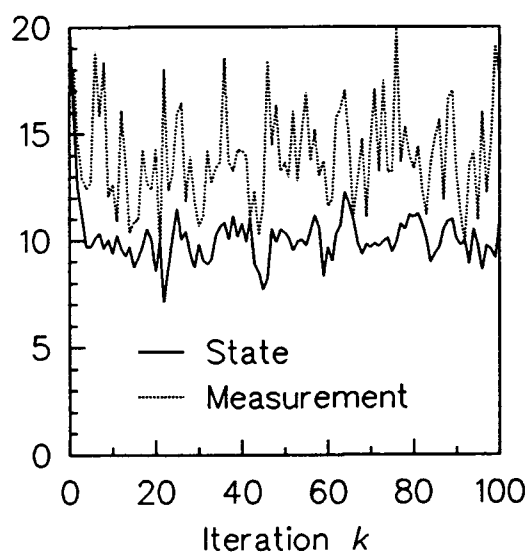
Figures 4a, b, and c show the weighting terms for each of the b_k bias terms. In order to lessen the noise of the weighting terms, a first-order lowpass filter

$$w_{k+1}^j = \lambda w_k^j + (1 - \lambda) w_{k+1}^j \quad (3.4.7)$$

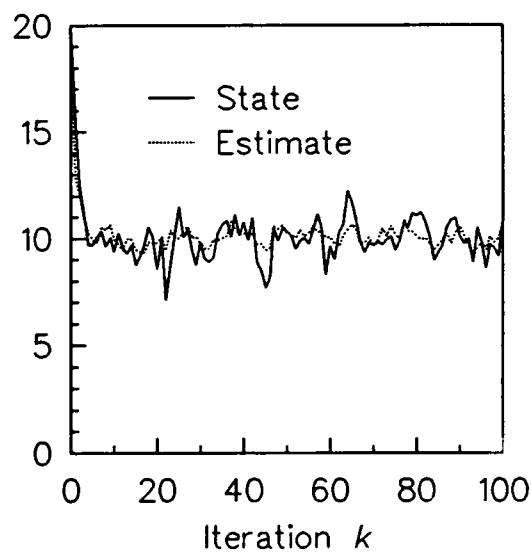
is used to smooth the weighting terms, where $\lambda = 0.7$. Figure 4d can be thought of as the overall measurement bias estimate \hat{b}_k due to the nonzero mean Gamma measurement noise. Using (3.3.35) and (3.3.50), this overall bias estimate is part of the overall state estimate (3.3.12) and is written in this case with $a_k = 0$ as

$$\hat{b}_k = \sum_{j=1}^M w_{k+1}^j b_k^j \quad (3.4.8)$$

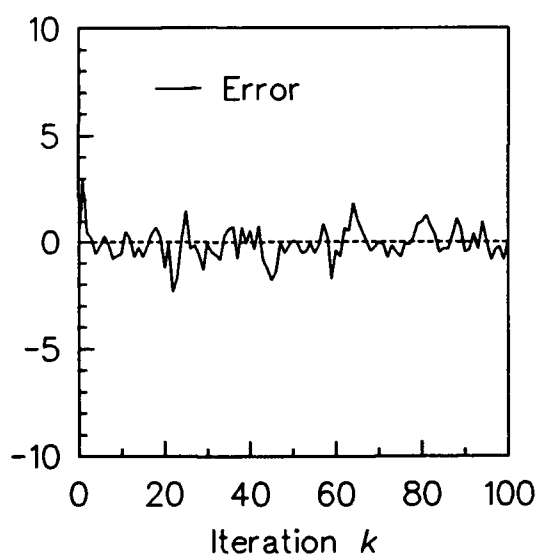
Note how this overall bias estimate approximately models the mean value of 4 of the Gamma measurement noise.



(a)

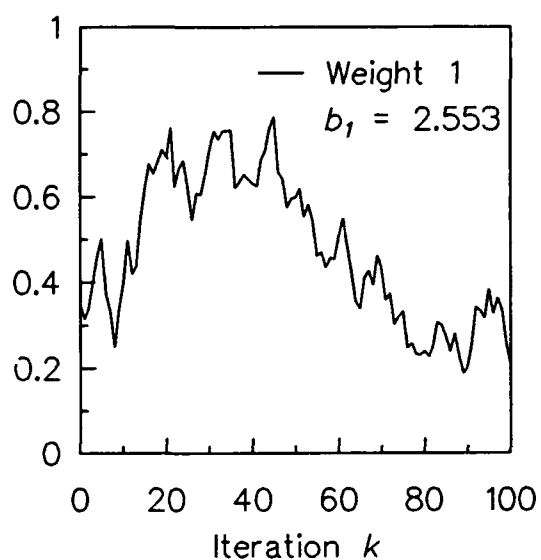


(b)

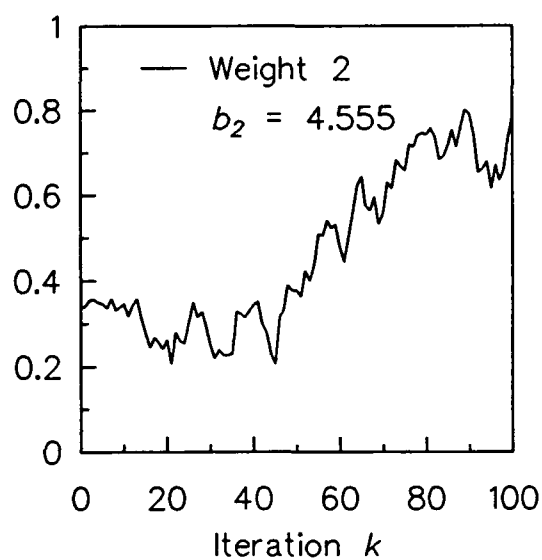


(c)

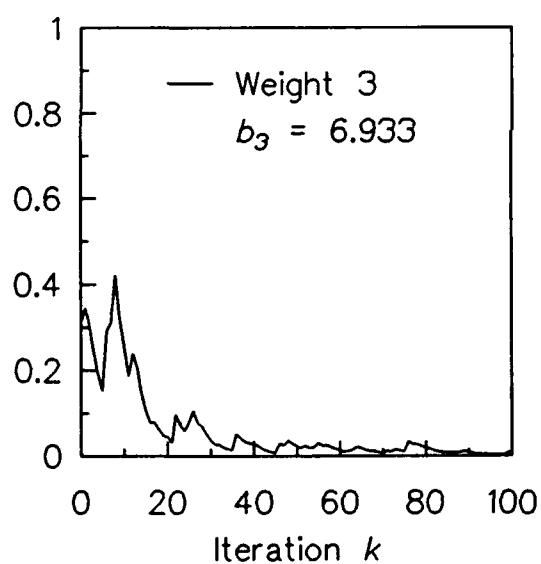
Figure 3. (a) Measurement and state simulation, (b) modified Gaussian sum estimate compared with state, (c) error = state - estimate.



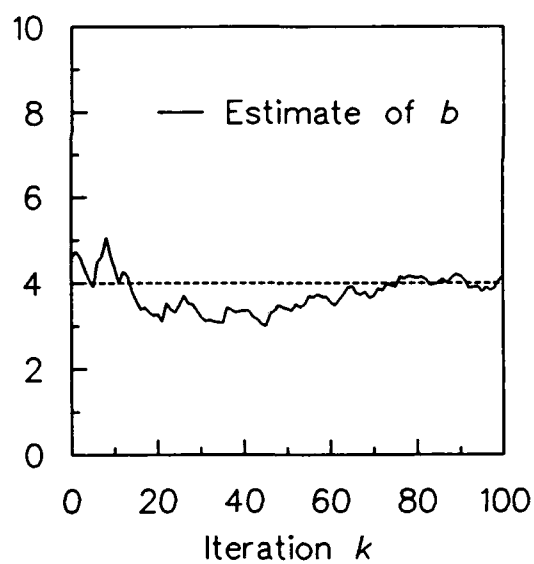
(a)



(b)



(c)



(d)

Figure 4. (a) Weight 1, (b) weight 2, (c) weight 3,
(d) overall measurement bias estimate.

3.5 Comparison of Gaussian Sum and Modified Gaussian Sum Algorithms

The Gaussian sum (GS) algorithm of Alspach [6], [7] and the modified Gaussian sum (MGS) algorithm developed in this present work exhibit several similarities. Among them are using a Gaussian sum density approximation to model the actual nonGaussian densities, a structure formed by a bank of Kalman filters operating in parallel, and weighting terms that are nonlinear functions of the measurement data. However, there are several key differences between the two algorithms that set them apart.

One difference is that the GS algorithm produces an exponentially increasing number of Gaussian terms at each iteration, while the MGS algorithm produces only a fixed number of Gaussian terms. As a result, the GS algorithm requires an exponentially increasing number of individual Kalman filters operating in parallel, which is impossible to implement for any practical purpose. The MGS algorithm avoids the increasing computational and storage requirements of the GS algorithm.

This leads to a second difference between the two. The GS algorithm is optimal while the MGS algorithm is suboptimal. Tugnait [55] and Raisch [56] point out that in order for the state estimate

$$\hat{x}_{k+1} = E[x_{k+1} | Z_{k+1}] = \sum_{i=1}^N \sum_{j=1}^M \hat{x}_{k+1}^{ij} p(a_i, b_j | Z_{k+1}) \quad (3.5.1)$$

to be optimal, the full exponentially increasing number of individual Kalman filters is required. This leads, in turn, to the condition that the individual ij^{th} estimate

$$\hat{x}_{k+1}^{ij} = \int_{-\infty}^{\infty} x_{k+1} p(x_{k+1} | Z_{k+1}, a_i, b_j) dx_{k+1} \quad (3.5.2)$$

is optimal only if the density $p(x_{k+1} | Z_{k+1}, a_i, b_j)$ is conditioned on the full measurement sequence $Z_{k+1} = \{z_1, z_2, \dots, z_{k+1}\}$ and the full switching-parameter bias term sequences $A_{k+1}^i = \{a_1, a_2, \dots, a_{k+1}\}^i$ and $B_{k+1}^j = \{b_1, b_2, \dots, b_{k+1}\}^j$ of present and past bias values for this particular ij^{th} estimate. Therefore, (3.5.2) would be written as

$$\hat{x}_{k+1}^{ij} = \int_{-\infty}^{\infty} x_{k+1} p(x_{k+1} | Z_{k+1}, A_{k+1}^i, B_{k+1}^j) dx_{k+1} \quad (3.5.3)$$

and the overall state estimate (3.5.1) would be

$$\hat{x}_{k+1} = \sum_{i=1}^{N_{k+1}} \sum_{j=1}^{M_{k+1}} \hat{x}_{k+1}^{ij} p(A_{k+1}^i, B_{k+1}^j | Z_{k+1}) \quad (3.5.4)$$

The weighting term $p(A_{k+1}^i, B_{k+1}^j | Z_{k+1})$ is also now a function of the full bias term sequences. The upper limits on the summations are no longer fixed constants, but are now functions of the time iteration. At each iteration, a is randomly selected from a fixed set of N discrete values. This produces N^{k+1} possible A_{k+1}^i sequences at iteration $k+1$, with i running from $1, 2, \dots, N^{k+1}$. Similarly, b is randomly selected from a fixed set of M discrete values, producing M^{k+1} possible B_{k+1}^j sequences, with j running from $1, 2, \dots, M^{k+1}$.

Rewriting the measurement sequence as $Z_{k+1} = \{z_{k+1}, Z_k\}$, and the bias term sequences as $A_{k+1}^i = \{a_{k+1}^i, A_k^i\}$ and $B_{k+1}^j = \{b_{k+1}^j, B_k^j\}$, the density of (3.5.3) is expanded as

$$p(x_{k+1} | z_{k+1}, Z_k, a_{k+1}^i, A_k^i, b_{k+1}^j, B_k^j) = \frac{N_1 N_2 N_3 N_4}{D_1 D_2 D_3} \quad (3.5.5)$$

where

$$N_1 = p(z_{k+1} | x_{k+1}, a_{k+1}^i, A_k^i, b_{k+1}^j, B_k^j, Z_k)$$

$$N_2 = p(a_{k+1}^i | x_{k+1}, A_k^i, b_{k+1}^j, B_k^j, Z_k)$$

$$N_3 = p(b_{k+1}^j | x_{k+1}, A_k^i, B_k^j, Z_k)$$

$$N_4 = p(x_{k+1} | A_k^i, B_k^j, Z_k)$$

$$D_1 = p(z_{k+1} | a_{k+1}^i, A_k^i, b_{k+1}^j, B_k^j, Z_k)$$

$$D_2 = p(a_{k+1}^i | A_k^i, b_{k+1}^j, B_k^j, Z_k)$$

$$D_3 = p(b_{k+1}^j | A_k^i, B_k^j, Z_k)$$

The N_2 term can be reduced to the Markov transition probability $N_2 = p(a_{k+1}^i | a_k^i)$ since the A_{k+1}^i is a semi-Markov sequence. The N_3 , D_2 , and D_3 can be reduced in a similar fashion, producing

$$N_2 = p(a_{k+1}^i | a_k^i), \quad N_3 = p(b_{k+1}^j | b_k^j)$$

$$D_2 = p(a_{k+1}^i | a_k^i), \quad D_3 = p(b_{k+1}^j | b_k^j)$$

Since $N_2 = D_2$ and $N_3 = D_3$, they each cancel out in (3.5.5) leaving

$$p(x_{k+1} | z_{k+1}, Z_k, a_{k+1}^i, A_k^i, b_{k+1}^j, B_k^j) =$$

$$\frac{p(z_{k+1} | x_{k+1}, a_{k+1}^i, A_k^i, b_{k+1}^j, B_k^j, Z_k) p(x_{k+1} | a_k^i, A_{k-1}^i, b_k^j, B_{k-1}^j, Z_k)}{p(z_{k+1} | a_{k+1}^i, A_k^i, b_{k+1}^j, B_k^j, Z_k)} \quad (3.5.6)$$

The density function of (3.5.2) used in the MGS algorithm is an approximation to (3.5.6), since it eliminates the past bias term sequences in (3.5.6). Using a similar development, the weighting term of (3.5.1) used in the MSG algorithm is an approximation to the weighting term of (3.5.4), since the past bias term sequences are eliminated here also. The MGS algorithm uses only the information contained

in the present bias values, thereby leading to a suboptimal state estimate (3.5.1).

Even though the GS algorithm produces an optimal state estimate, it can never be found in practice since the number of GS terms increases exponentially from one iteration to the next according to (3.1.15) and (3.1.22), combined below

$$\xi'_{k+1} = q_k r_k \xi'_k \quad (3.5.7)$$

In order to reduce this growing number to some prescribed fixed number Ξ' at each iteration, Alspach [6] proposes a suboptimal estimate based on observed mechanisms that allow terms to be neglected or combined. At iteration k , he drops weighting terms (3.1.16) and (3.1.23) that fall below some prescribed threshold. He also combines several GS terms of (3.1.11) into one term if their means (3.1.12) and variances (3.1.13) have become approximately equal. These two operations are used until only $\xi'_k = \Xi'$ significant terms remain. Then, at iteration $k+1$, the number of GS terms grows again to the larger number ξ'_{k+1} according to (3.5.7), is reduced by eliminating and combining terms until $\xi'_{k+1} = \Xi'$, and the cycle repeats for all subsequent iterations. This method results in a substantial reduction in the number of GS terms with a negligible effect on the $p(x_{k+1} | Z_{k+1})$ density approximation.

On the surface, it appears that Alspach's suboptimal Gaussian sum (SGS) algorithm can achieve performance equal to that of the MGS algorithm, since only a fixed number of terms is used to produce the state estimate. However, in order for the SGS algorithm to produce the fixed number of terms Ξ' , the larger number of terms $q_k r_k \Xi'$ must first be generated, requiring $q_k r_k \Xi'$ parallel filters. The MGS algorithm requires only Ξ' filters at each iteration to produce a Ξ' term state estimate. The computational savings of the MGS algorithm is considerable. Even

if the plant and measurement noise processes (3.1.5) and (3.1.6) were approximated using the minimum number of two terms each ($q_k = r_k = 2$), the SGS algorithm would require a number of filters $q_k r_k = 4$ times greater than the MGS algorithm.

A third difference is that the GS algorithm must always use time-varying Kalman gains while the MGS algorithm allows the use of steady-state Kalman gains. The growing memory problem of the GS algorithm prevents steady-state Kalman gains to be calculated off-line before running the state estimator. The SGS algorithm must also use time-varying gains because the elimination and combination of terms is performed at each iteration. Therefore, the Kalman gains could never be computed in advance. The MGS algorithm uses the same fixed number of filters at each iteration. With constant covariances Q and R of the plant noise (3.3.2) and measurement noise (3.3.4) used in the gain equations for each ij^{th} filter, steady-state Kalman gains could be calculated off-line. Even if Q and R were different for each ij^{th} filter, they would then be modeled as semi-Markov processes, with values randomly selected from fixed sets of discrete values. Since all the covariance values would be known in advance, steady-state Kalman gains could again be calculated off-line.

A fourth difference between the two algorithms is the use of Markov transition probabilities in the weighting term of the MGS algorithm, restated here for convenience

$$w_{k+1}^{ij} = C_{k+1}^o N[\tilde{z}_{k+1}^{ij}, R_{k+1}] \sum_{\alpha=1}^N \sum_{\beta=1}^M \theta_a^{i\alpha} \theta_b^{j\beta} w_k^{\alpha\beta} \quad (3.5.8)$$

The weighting term of the GS algorithm at iteration $k+1$ is

$$\alpha_{k+1,j} = \frac{\alpha'_{k+1,i} \gamma_{k+1,m} W_{k+1,j}}{\sum_{i=1}^{\xi'_{k+1}} \alpha'_{k+1,i} \gamma_{k+1,m} W_{k+1,j}} \quad (3.5.9)$$

Define the scale factor as

$$\lambda_{k+1}^o = \frac{1}{\xi_{k+1}' \sum_{i=1} \alpha_{k+1,i}' \gamma_{k+1,m} W_{k+1,j}} \quad (3.5.10)$$

Using the definition of the $\alpha_{k+1,i}'$ term

$$\alpha_{k+1,i}' = \alpha_{k_l} \beta_{k_n} \quad (3.5.11)$$

and (3.5.10) in (3.5.9) results in

$$\alpha_{k+1,j} = \lambda_{k+1}^o W_{k+1,j} [\gamma_{k+1,m} \beta_{k_n}] \alpha_{k_l} \quad (3.5.12)$$

The GS weighting term (3.5.12) now has a similar recursive form as the MGS weighting term (3.5.8), with λ_{k+1}^o and C_{k+1}^o being the respective scale factors, and $W_{k+1,j}$ and $N[\tilde{z}_{k+1}^{ij}, R_{k+1}]$ being the respective Gaussian density functions based on the measurement data. The main difference between the two weighting terms is the Markov transition probabilities in (3.5.8) and the $\gamma_{k+1,m} \beta_{k_n}$ factor in (3.5.12). The Markov probabilities govern the chances that a bias term is going to switch from one value to another. The $\gamma_{k+1,m} \beta_{k_n}$ factor is formed from the weighting terms of the Gaussian sum approximations of the nonGaussian densities (3.1.5) and (3.1.6), and has no meaning similar to the Markov probabilities. Of course, the current number of $\alpha_{k+1,j}$ terms is larger than the previous number of α_{k_l} terms at the each iteration, while the number of w_{k+1}^{ij} terms remains fixed at each iteration.

4.0 APPLICATION TO A MODAL SYSTEM

The modified Gaussian sum (MGS) adaptive filtering technique of chapter 3 is now applied to a modal system problem. Large, heavily damped structures modeled by a plant with complex modes are considered in [1]. The input to the plant is unknown, but its frequency characteristics are known and the input is always present. The nonGaussian nature of the input signal is modeled using the signal model of chapter 3, and the MGS adaptive filter for this system is developed. A simulation example using the MGS adaptive filter is given and the results are compared to those produced from an augmented Kalman filter based on a system model assuming a narrowband Gaussian input signal. A necessary condition for effective estimation is derived. Alternate estimation procedures are developed to compensate for situations when this condition is not met. Several configurations are simulated and their performance results are analyzed and compared.

4.1 Modal System and Filter Development

The MGS algorithm is applied to a modal system with a nonGaussian input

signal and Gaussian plant and measurement noise. The theoretical foundation for this system, which serves as a model for large, heavily damped structures, is found in [1] and briefly outlined in appendix A. Using a zero-order-hold model, the resulting discrete-time system equations with added noise terms are

$$x_{k+1} = \Phi x_k + \Gamma u_k + \Psi w_k \quad (4.1.1)$$

$$z_k = H x_k + D u_k + v_k \quad (4.1.2)$$

$$u_k = b_k + n_k \quad (4.1.3)$$

where

x_{k+1} is the state vector

z_k is the measurement vector

w_k is a zero mean white Gaussian plant noise process with covariance Q

v_k is a zero mean white Gaussian measurement noise process with covariance R

Φ, Γ, Ψ, H, D are the respective constant transition matrices

u_k is the vector Gaussian sum (GS) signal model of the actual nonGaussian input signal, comprised of semi-Markov bias vector b_k , and zero mean white Gaussian noise n_k with semi-Markov covariance S_k

and the random quantities x_k , w_k , v_k , b_k , and n_k are assumed to be mutually statistically independent.

There are two important differences between the modal system of (4.1.1) - (4.1.3) and the general system of chapter 3, (3.3.1) - (3.3.4). The first difference is that in the modal system, the signal u_k is an input not only to the plant, but is also

fed through to the measurement z_k . Inserting (4.1.3) into (4.1.1) and (4.1.2) produces modified plant and measurement equations

$$x_{k+1} = \Phi x_k + \Gamma b_k + \Psi w'_k \quad (4.1.4)$$

$$z_k = H x_k + D b_k + v'_k \quad (4.1.5)$$

where

$$w'_k = \Gamma n_k + \Psi w_k \quad (4.1.6)$$

$$v'_k = D n_k + v_k \quad (4.1.7)$$

Because n_k appears in both (4.1.6) and (4.1.7), w'_k and v'_k are correlated Gaussian random processes. At first glance, this correlation may seem to be an obstacle in developing the Kalman filter equations for this model, since the Kalman filter development requires that the plant and measurement noise processes be uncorrelated. However, what is really necessary in this situation is for w'_{k-1} and v'_k to be uncorrelated [57]. Rewriting (4.1.4) for x_k gives

$$x_k = \Phi x_{k-1} + \Gamma b_{k-1} + \Psi w'_{k-1} \quad (4.1.8)$$

and substituting (4.1.8) into (4.1.5) gives

$$z_k = H(\Phi x_{k-1} + \Gamma b_{k-1} + \Psi w'_{k-1}) + D b_k + v'_k \quad (4.1.9)$$

thereby showing that any covariance calculations involving (4.1.9) will involve the correlation between w'_{k-1} and v'_k , and not between w'_k and v'_k . The correlation between w'_{k-1} and v'_k is given by

$$\begin{aligned}
E[w'_{k-1}v_k^T] &= E[(\Gamma n_{k-1} + \Psi w_{k-1})(Dn_k + v_k)^T] \\
&= \Gamma E[n_{k-1}n_k^T]D^T + \Gamma E[n_{k-1}v_k^T] + \Psi E[w_{k-1}n_k^T]D^T + \Psi E[w_{k-1}v_k^T] \quad (4.1.10) \\
&= 0
\end{aligned}$$

since w_k , v_k , and n_k are assumed to be mutually statistically independent.

The second difference is that the covariance S_k of the GS signal model is no longer constant, but can vary with time. The bias vector b_k and covariance S_k are both semi-Markov processes with state transitions governed by a single Markov transition probability matrix. That is, values of b_k and S_k will be paired together and will randomly switch from one pair to another. The switching covariance can be thought of as changing the power of the Gaussian process.

As in chapter 3, the optimal estimate \hat{x}_{k+1} of the state vector is found by minimizing the mean-square error between x_{k+1} and \hat{x}_{k+1} . This results in the conditional mean estimate

$$\hat{x}_{k+1} = E[x_{k+1} | Z_{k+1}] = \int_{-\infty}^{\infty} x_{k+1} p(x_{k+1} | Z_{k+1}) dx_{k+1} \quad (4.1.11)$$

where Z_{k+1} is the current measurement sequence $\{z_1, z_2, \dots, z_{k+1}\}$. The conditional density function of (4.1.11) can be written as the ratio of the corresponding joint and marginal densities

$$p(x_{k+1} | Z_{k+1}) = \frac{p(x_{k+1}, Z_{k+1})}{p(Z_{k+1})} \quad (4.1.12)$$

The bias vector b_k and covariance matrix S_k are explicitly brought into (4.1.12) by considering the joint density $p(x_{k+1}, Z_{k+1})$ to be a marginal density found from $p(x_{k+1}, Z_{k+1}, b_k = b_i, S_k = S_i)$ by summing over the b_k and S_k terms

$$p(x_{k+1} | Z_{k+1}) = \frac{\sum_{i=1}^M p(x_{k+1}, Z_{k+1}, b_k = b_i, S_k = S_i)}{p(Z_{k+1})} \quad (4.1.13)$$

After some additional algebraic manipulation, the optimal estimate of (4.1.11) becomes

$$\hat{x}_{k+1} = \sum_{i=1}^M \hat{x}_{k+1}^i p(b_i, S_i | Z_{k+1}) \quad (4.1.14)$$

where \hat{x}_{k+1}^i is the conditional mean estimate of x_{k+1} given that $b_k = b_i$ and $S_k = S_i$, denoted by

$$\hat{x}_{k+1}^i = \int_{-\infty}^{\infty} x_{k+1} p(x_{k+1} | Z_{k+1}, b_i, S_i) dx_{k+1} \quad (4.1.15)$$

and the weighting factor $p(b_i, S_i | Z_{k+1})$ is the probability that $b_k = b_i$ and $S_k = S_i$ given the current measurement sequence. The overall estimate of (4.1.14) is then a weighted sum of M individual estimates, each based on a particular pair of parameters b_i and S_i .

Each individual estimate (4.1.15) is found by a modified Kalman filter. Using the same method in chapter 3, the Kalman filter equation for (4.1.15) is

$$\hat{x}_{k+1}^i = \Phi \hat{x}_k^i + \Gamma b_i + K_{k+1}^i [z_{k+1} - D b_i - H(\Phi \hat{x}_k^i + \Gamma b_i)] \quad (4.1.16)$$

with covariance and gain equations

$$M_{k+1}^i = \Phi P_k^i \Phi^T + \Gamma S_i \Gamma^T + \Psi Q \Psi^T \quad (4.1.17)$$

$$K_{k+1}^i = M_{k+1}^i H^T (H M_{k+1}^i H^T + D S_i D^T + R)^{-1} \quad (4.1.18)$$

$$P_{k+1}^i = (I - K_{k+1}^i H) M_{k+1}^i \quad (4.1.19)$$

Note that (4.1.17) - (4.1.19) are now dependent on S_i so that an individual set of these equations has to be computed for each i^{th} estimator (4.1.16), whereas in chapter 3 only one set was computed and used for all individual estimators.

As in chapter 3, the next step in producing the overall estimate (4.1.14) is to find the weighting term $p(b_i, S_i | Z_{k+1})$. Using Bayes' theorem, and writing Z_{k+1} as $\{Z_k, z_{k+1}\}$, the weighting term becomes

$$p(b_i, S_i | Z_{k+1}) = \frac{p(z_{k+1} | b_i, S_i, Z_k) p(b_i, S_i | Z_k)}{p(z_{k+1} | Z_k)} \quad (4.1.20)$$

The denominator term is independent of i and is a scale factor that ensures that the sum of the weights (4.1.20) at each iteration is equal to one. The first term of the numerator can be modeled as a Gaussian density if the bias and covariance terms switch slowly compared to the sample interval k . This is the same assumption made in chapter 3. This conditional density is

$$p(z_{k+1} | b_i, S_i, Z_k) = N[H(\Phi \hat{x}_k^i + \Gamma b_i) + D b_i, H M_{k+1}^i H^T + D S_i D^T + R] \quad (4.1.21)$$

The mean and covariance of (4.1.21) are readily available from the Kalman filter equation (4.1.16) and the Kalman gain (4.1.18).

The second term of the numerator can be rewritten to explicitly include the sample time for the bias and covariance terms

$$p(b_i, S_i | Z_k) = p(b_{k+1} = b_i, S_{k+1} = S_i | Z_k) \quad (4.1.22)$$

Using Bayes' rule and the definition of marginal densities, (4.1.22) is expanded into

$$p(b_i, S_i | Z_k) = \sum_{j=1}^M p(b_{k+1}=b_i, S_{k+1}=S_i | b_k=b_j, S_k=S_j) p(b_k=b_j, S_k=S_j | Z_k) \quad (4.1.23)$$

As in chapter 3, the following notation is used to express the terms of (4.1.23)

$$\theta_{bS}^{ij} = p(b_{k+1}=b_i, S_{k+1}=S_i | b_k=b_j, S_k=S_j) \quad (4.1.24)$$

$$w_k^j = p(b_k=b_j, S_k=S_j | Z_k) \quad (4.1.25)$$

The term (4.1.24) is a Markov transition probability [22]; that is, θ_{bS}^{ij} is the conditional probability that $b=b_i$ and $S=S_i$ at time $k+1$, given that $b=b_j$ and $S=S_j$ at time k . The term (4.1.25) is of the same form as (4.1.20) and is just the previous weighting term at the previous time interval k .

Combining (4.1.21), (4.1.24), and (4.1.25), the weighting term (4.1.20) is written as

$$w_{k+1}^i = p(b_i, S_i | Z_{k+1}) = C_{k+1}^o p(z_{k+1} | b_i, S_i, Z_k) \sum_{j=1}^M \theta_{bS}^{ij} w_k^j \quad (4.1.26)$$

where C_{k+1}^o is a scale factor determined at each time interval such that the sum of all the weighting terms (4.1.26) is equal to one.

The equations for the modified Gaussian sum adaptive filter for the system of (4.1.1) - (4.1.3) are summarized in Table 3. The structure of the overall adaptive filter is again a bank of Kalman filters operating in parallel, with each individual estimate multiplied by its own corresponding weighting term. The i^{th} estimator based on the bias and covariance pair that most closely matches the actual bias and covariance pair of the modeled system will have a corresponding weighting

term that tends closer to one, while the weights of the other mismatched estimators will tend towards zero. A block diagram of the MGS adaptive filter for this modal system is shown in Figure 5.

Table 3.

Modified Gaussian sum adaptive filter equations for a modal system.

System:

$$x_{k+1} = \Phi x_k + \Gamma u_k + \Psi w_k$$

$$z_k = H x_k + D u_k + v_k$$

$$u_k = b_k + n_k$$

Overall estimate:

$$\hat{x}_{k+1} = \sum_{i=1}^M \hat{x}_{k+1}^i p(b_i, S_i | Z_{k+1})$$

Kalman filter equation:

$$\hat{x}_{k+1}^i = \Phi \hat{x}_k^i + \Gamma b_i + K_{k+1}^i [z_{k+1} - D b_i - H(\Phi \hat{x}_k^i + \Gamma b_i)]$$

Kalman gain equations:

$$M_{k+1}^i = \Phi P_k^i \Phi^T + \Gamma S_i \Gamma^T + \Psi Q \Psi^T$$

$$K_{k+1}^i = M_{k+1}^i H^T (H M_{k+1}^i H^T + D S_i D^T + R)^{-1}$$

$$P_{k+1}^i = (I - K_{k+1}^i H) M_{k+1}^i$$

Weighting term:

$$w_{k+1}^i = p(b_i, S_i | Z_{k+1}) = C_{k+1}^o p(z_{k+1} | b_i, S_i, Z_k) \sum_{j=1}^M \theta_{bS}^{ij} w_k^j$$

with

$$p(z_{k+1} | b_i, S_i, Z_k) = N[H(\Phi \hat{x}_k^i + \Gamma b_i) + D b_i, H M_{k+1}^i H^T + D S_i D^T + R]$$

$$\theta_{bS}^{ij} = p(b_{k+1} = b_i, S_{k+1} = S_i | b_k = b_j, S_k = S_j)$$

$$w_k^j = p(b_k = b_j, S_k = S_j | Z_k)$$

and scale factor C_{k+1}^o such that $\sum_{i=1}^M w_{k+1}^i = 1$

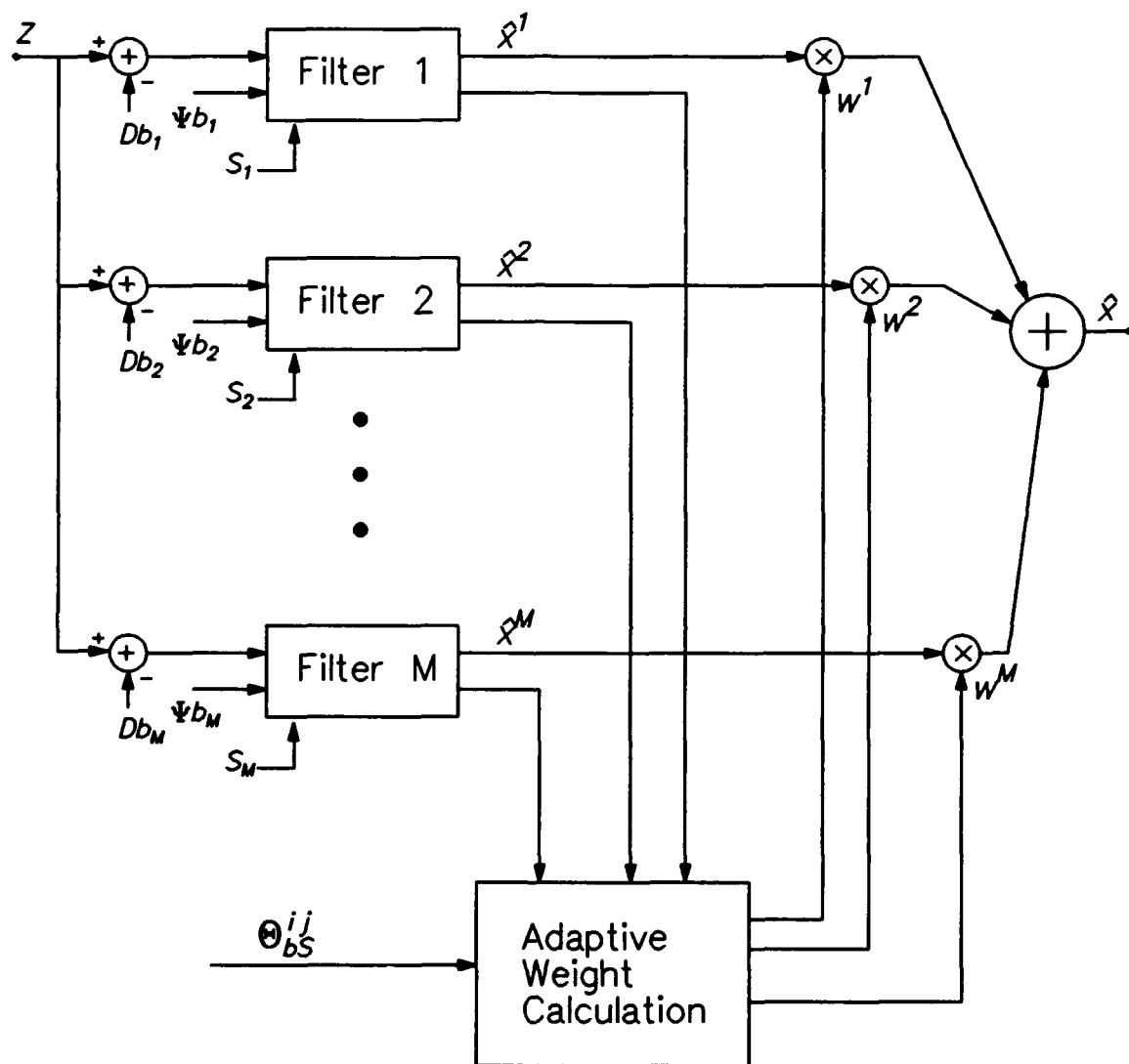


Figure 5. Modified Gaussian sum adaptive filter structure for a modal system.

4.2 Simulation Example

An example applying the MGS algorithm to a modal system with a nonGaussian input signal is now presented. The results are compared to those produced from an augmented system model assuming that the input signal is Gaussian. A second-order system is used, modeled by the transfer function

$$G(s) = \frac{s^2 + cs + d}{s^2 + as + b} \quad (4.2.1)$$

and in continuous-time state space form as

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (4.2.2)$$

$$y(t) = Cx(t) + Du(t) \quad (4.2.3)$$

The system matrices are defined by

$$A = \begin{bmatrix} 0 & 1 \\ -b & -a \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad C = \begin{bmatrix} d-b & c-a \end{bmatrix} \quad D = \begin{bmatrix} 1 \end{bmatrix} \quad (4.2.4)$$

where the resonant frequency of the system is $f_o = 20$ Hz and the damping coefficient is $\zeta_o = 0.1$ so that

$$a = 2\zeta_o(2\pi f_o) \quad b = (2\pi f_o)^2 \quad c = 5a \quad d = 5b \quad (4.2.5)$$

Using a zero-order-hold model with a sample time of $T = \frac{1}{60}$ second, the equivalent discrete-time system with added noise terms becomes

$$x_{k+1} = \Phi x_k + \Gamma u_k + \Psi w_k \quad (4.2.6)$$

$$z_k = H x_k + D u_k + v_k \quad (4.2.7)$$

where

$$\Phi = \begin{bmatrix} -0.3271 & 0.0057 \\ -89.2411 & -0.4691 \end{bmatrix} \quad \Gamma = \begin{bmatrix} 0.0001 \\ 0.0057 \end{bmatrix} \quad \Psi = \begin{bmatrix} 1 \\ 100 \end{bmatrix} \quad (4.2.8)$$

$$H = \begin{bmatrix} 63165 & 100.53 \end{bmatrix} \quad D = \begin{bmatrix} 1 \end{bmatrix}$$

The plant noise w_k is zero-mean white Gaussian with covariance $Q = 10^{-10}$, and is uncorrelated with measurement noise v_k , which is also zero-mean white Gaussian with covariance $R = 10^{-2}$.

The nonGaussian input signal is a stochastic FM signal generated by frequency modulating a sinusoid with a Gaussian process. The form of the signal is

$$u(t) = A_u \sin(2\pi f_u t + k_u \int_0^t m(\tau) d\tau) \quad (4.2.9)$$

where the amplitude is $A_u = 5$, the carrier frequency is $f_u = 2$ Hz, the modulation index is $k_u = 10$, and $m(t)$ is zero-mean white Gaussian noise with variance 1. Figure 6a shows the FM signal $u(t)$ for 1 second. The probability density function of $u(t)$ has the form [58]

$$p(u) = \begin{cases} \frac{1}{\pi \sqrt{A_u^2 - u^2}} & |u| < A_u \\ 0 & |u| \geq A_u \end{cases} \quad (4.2.10)$$

Figure 6b shows a plot of the density function of (4.2.10), while Figure 6c shows a normalized histogram of a 10 second sample of $u(t)$. Note how closely the

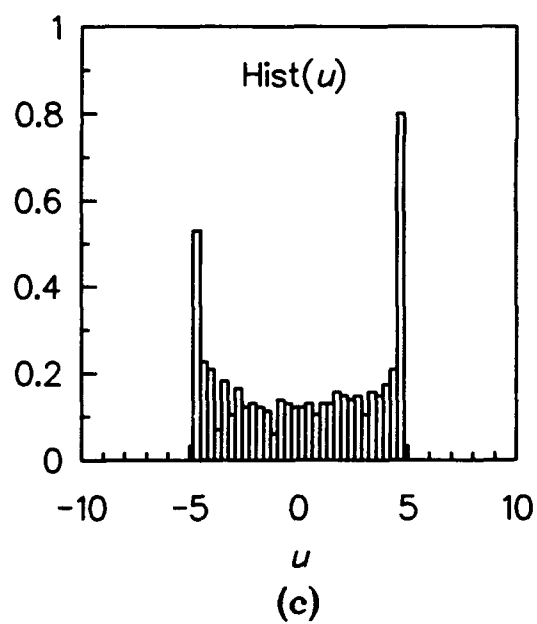
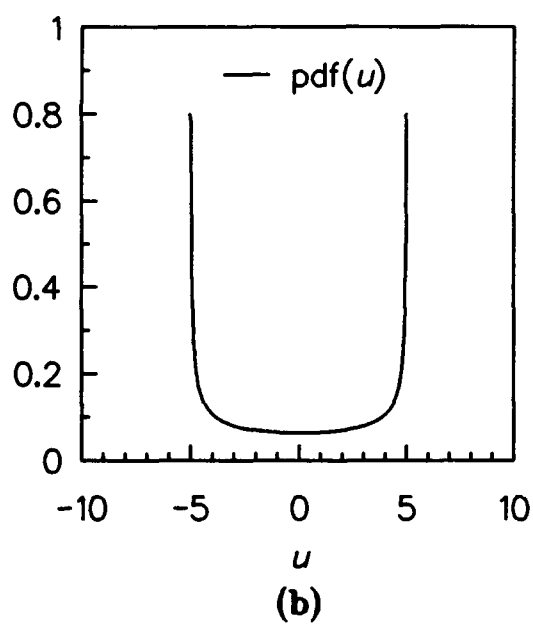
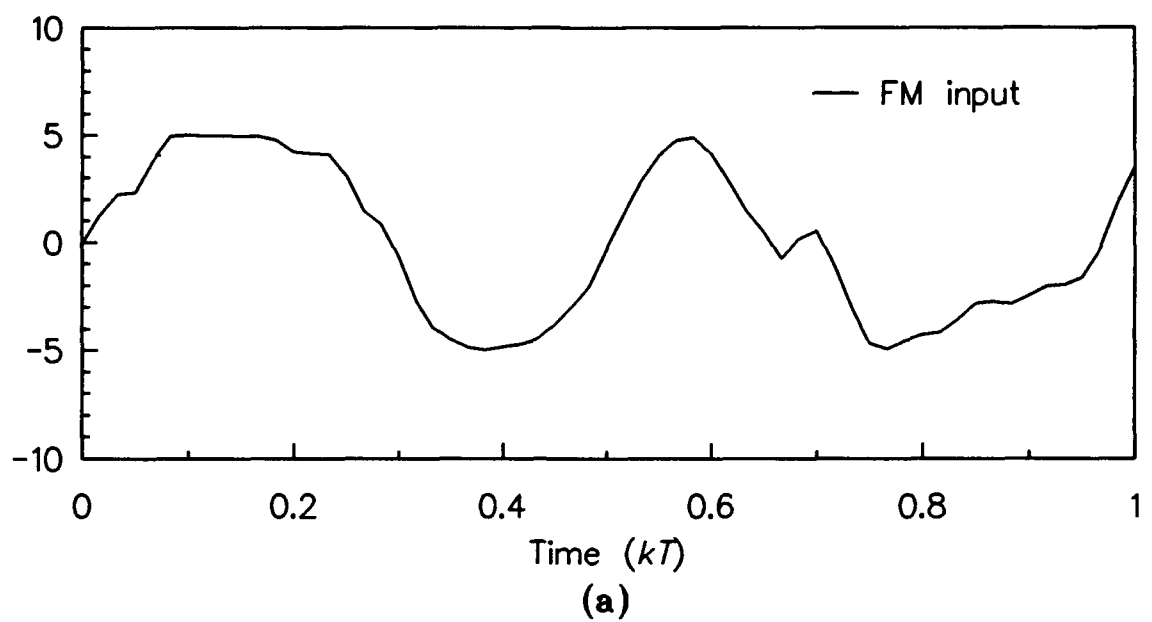


Figure 6. (a) Stochastic FM signal $u(t)$, (b) probability density function of $u(t)$, (c) normalized histogram of $u(t)$.

histogram matches the actual density function.

The nonGaussian input signal $u(t)$ of (4.2.9) will be modeled in discrete-time using the Gaussian sum (GS) signal model described in section 4.1

$$u_k = b_k + n_k \quad (4.2.11)$$

with GS density approximation described in chapter 3

$$p(u) = \sum_{i=1}^M P_i N[b_i, S_i] \quad (4.2.12)$$

Figure 7 compares the nonGaussian density function (4.2.10) of the stochastic FM input signal with a three-term GS density approximation (4.2.12) minimizing the L^2 norm (3.2.11). Table 4 lists the values of P_i , b_i , and S_i used in the Gaussian sum approximation and the MGS adaptive filter.

Table 4.
Gaussian sum P_i , b_i , and S_i values.

| i | P_i | b_i | S_i |
|-----|-------|-------|-------|
| 1 | 0.1 | -5.0 | 0.04 |
| 2 | 0.1 | 5.0 | 0.04 |
| 3 | 0.8 | 0.0 | 4.28 |

It is assumed that the nonGaussian input signal $u(t)$ is much larger than both the Gaussian plant noise w_k and the Gaussian measurement noise v_k . However, allowing the power of $u(t)$ to be larger than the power (covariance Q) of w_k and

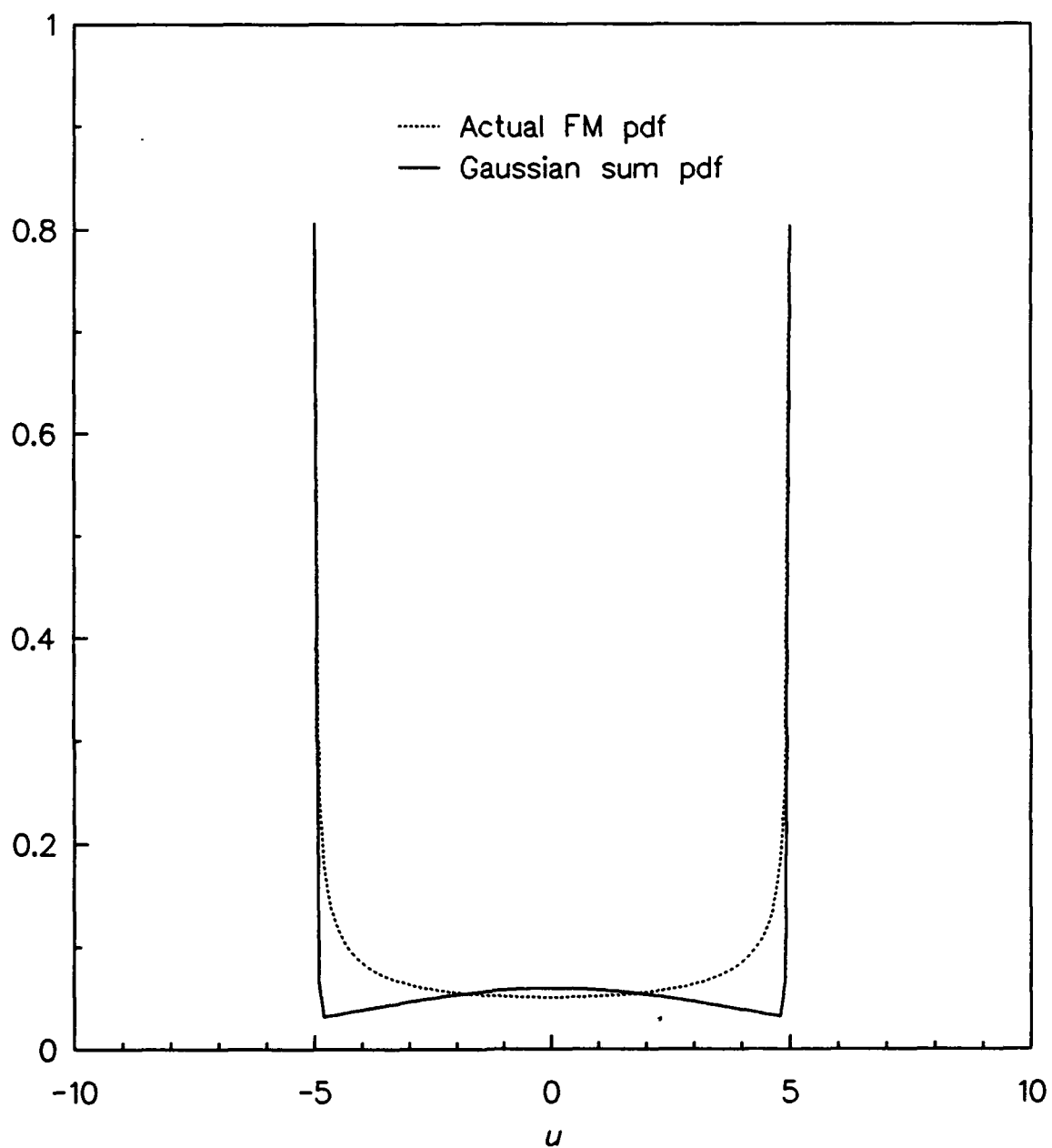


Figure 7. Stochastic FM signal and Gaussian sum probability density functions.

power (covariance R) of v_k is not enough to ensure good MGS simulation results. The nonGaussian input signal $u(t)$ is modeled by u_k (4.2.11), which will take on different b_i, S_i pairs at various times. The lowest value of S_i from Table 4 is $S_i = 0.04$. From covariance equation (4.1.18) and Kalman gain (4.1.19), repeated here for convenience

$$M_{k+1}^i = \Phi P_k^i \Phi^T + \Gamma S_i \Gamma^T + \Psi Q \Psi^T$$

$$K_{k+1}^i = M_{k+1}^i H^T (H M_{k+1}^i H^T + D S_i D^T + R)^{-1}$$

it is apparent that for good MGS simulation results, covariances Q and R must be selected such that the elements of $\Gamma S_i \Gamma^T$ are larger than the elements of $\Psi Q \Psi^T$, and the elements of $D S_i D^T$ are larger than the elements of R . This specification is met using $S_i = 0.04$, $Q = 10^{-10}$, $R = 10^{-2}$, and Γ, Ψ , and D from (4.2.8), giving

$$\Gamma S_i \Gamma^T = \begin{bmatrix} 2.8 \times 10^{-10} & 1.9 \times 10^{-8} \\ 1.9 \times 10^{-8} & 1.3 \times 10^{-6} \end{bmatrix} \quad \Psi Q \Psi^T = \begin{bmatrix} 1.0 \times 10^{-10} & 1.0 \times 10^{-8} \\ 1.0 \times 10^{-8} & 1.0 \times 10^{-6} \end{bmatrix}$$

$$D S_i D^T = 0.04 \quad R = 0.01$$

For comparison purposes, the nonGaussian input signal $u(t)$ of (4.2.9) will also be modeled as a narrowband Gaussian signal, generated by passing white Gaussian noise through a linear, time-invariant filter. This shaping filter is designed using the frequency characteristics of $u(t)$. Figure 8 shows the power spectral density of $u(t)$ and the power spectral density of a second-order shaping filter, modeled by the transfer function

$$G_{nb}(s) = \frac{N_{nb}s}{s^2 + a_{nb}s + b_{nb}} \quad (4.2.13)$$

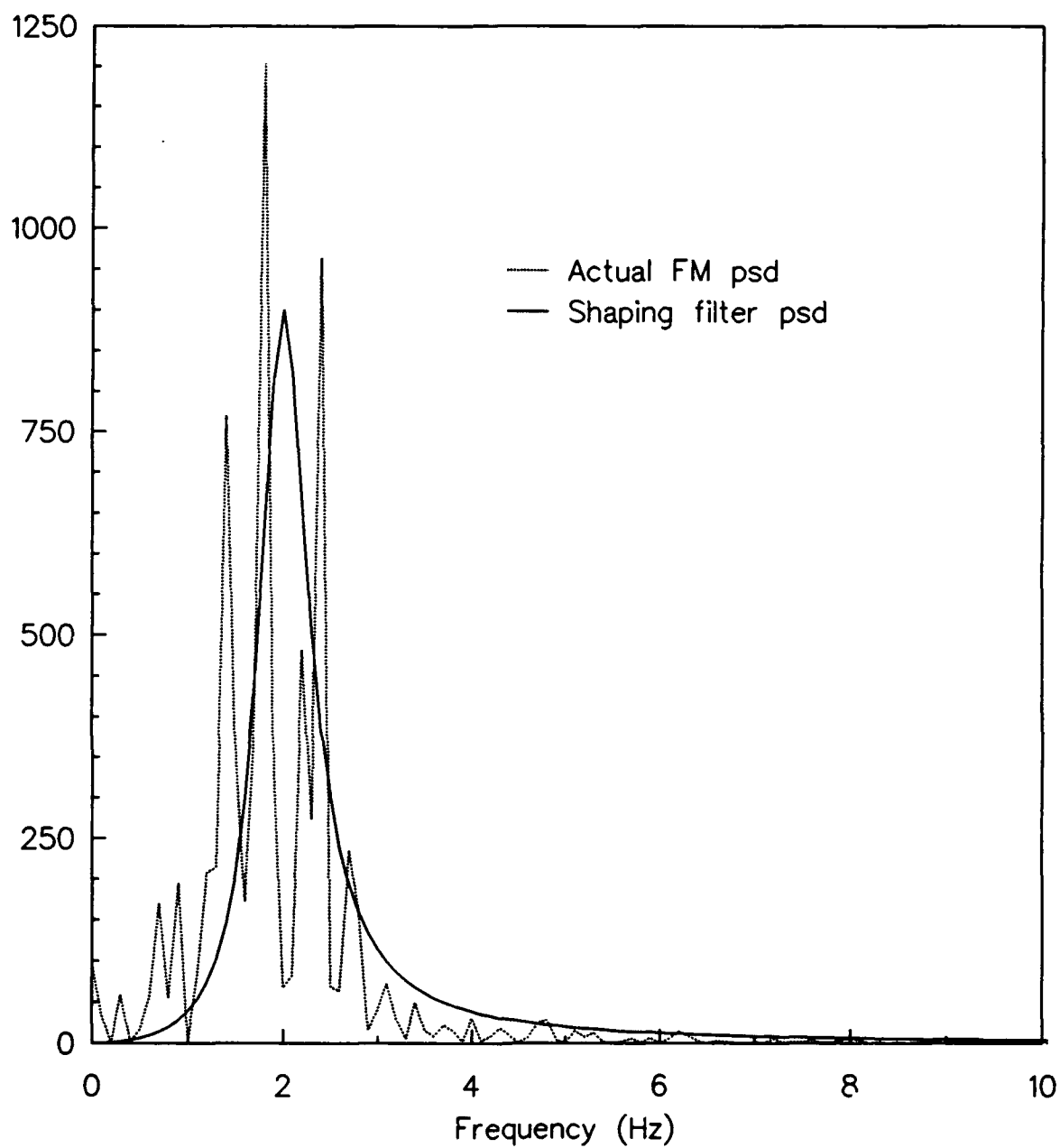


Figure 8. Stochastic FM signal and shaping filter power spectral densities.

and in continuous-time state space form as

$$\dot{x}_{nb}(t) = A_{nb}x_{nb}(t) + B_{nb}w_{nb}(t) \quad (4.2.14)$$

$$u_{nb}(t) = C_{nb}x_{nb}(t) \quad (4.2.15)$$

The system matrices are defined by

$$A_{nb} = \begin{bmatrix} 0 & 1 \\ -b_{nb} & -a_{nb} \end{bmatrix} \quad B_{nb} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad C_{nb} = \begin{bmatrix} 0 & 1 \end{bmatrix} \quad (4.2.16)$$

with resonant frequency $f_{nb} = 2$ Hz and damping coefficient $\zeta_{nb} = \frac{1}{2\pi}$ such that

$$a_{nb} = 2\zeta_{nb}(2\pi f_{nb}) = 4 \quad b_{nb} = (2\pi f_{nb})^2 = 157.91 \quad (4.2.17)$$

and w_{nb} is zero-mean white Gaussian noise with variance $N_{nb}^2 = (120)^2$.

The Gaussian input model $u_{nb}(t)$ of (4.2.15) is now assumed to model the stochastic FM input $u(t)$ of (4.2.9) by augmenting the system equations (4.2.2) and (4.2.3) with the shaping filter equations (4.2.14) and (4.2.15), giving

$$\dot{x}_a(t) = A_a x_a(t) + B_a w_{nb}(t) \quad (4.2.18)$$

$$y_a(t) = C_a x_a(t) \quad (4.2.19)$$

The system matrices of (4.2.18) and (4.2.19) are defined by

$$x_a(t) = \begin{bmatrix} x(t) \\ x_{nb}(t) \end{bmatrix} \quad A_a = \begin{bmatrix} A & BC_{nb} \\ 0 & A_{nb} \end{bmatrix} \quad B_a = \begin{bmatrix} 0 \\ B_{nb} \end{bmatrix} \quad C_a = \begin{bmatrix} C & DC_{nb} \end{bmatrix} \quad (4.2.20)$$

Using a zero-order-hold model with a sample time of $T = \frac{1}{60}$ second, the equivalent

discrete-time augmented system with added noise terms becomes

$$x_{k+1}^a = \Phi_a x_k^a + w_k^{nb} + \Psi_a w_k \quad (4.2.21)$$

$$z_k^a = H_a x_k^a + v_k \quad (4.2.22)$$

where

$$\Phi_a = \begin{bmatrix} -0.3271 & 0.0057 & -0.00009 & 0.00008 \\ -89.2411 & -0.4691 & -0.0129 & 0.00524 \\ 0 & 0 & 0.9786 & 0.0160 \\ 0 & 0 & -2.5275 & 0.9146 \end{bmatrix} \quad \Psi_a = \begin{bmatrix} 1 \\ 100 \\ 0 \\ 0 \end{bmatrix} \quad (4.2.23)$$

$$H_a = \begin{bmatrix} 63165 & 100.53 & 0 & 1 \end{bmatrix}$$

The first plant noise term w_k^{nb} is the discrete-time version of the continuous-time plant noise $w_{nb}(t)$ of (4.2.14), with new covariance found from [5] as

$$Q_{nb} = \int_0^T e^{A_a \tau} B_a N_{nb}^2 B_a^T e^{A_a^T \tau} d\tau$$

or

$$Q_{nb} = \begin{bmatrix} 0.00002 & 0.0024 & 0.0047 & 0.0374 \\ 0.0024 & 0.0329 & 0.0564 & 55.85 \\ 0.0047 & 0.0564 & 1.048 & 92.22 \\ 0.0374 & 55.85 & 92.22 & 11194 \end{bmatrix} \quad (4.2.24)$$

The second plant noise term w_k and the measurement noise v_k are the same as those found in (4.2.6) and (4.2.7). The three processes w_k^{nb} , w_k , and v_k are assumed to be mutually statistically independent.

The Kalman filter based on the augmented system (4.2.21) and (4.2.22), is given by

$$\hat{x}_{k+1}^a = \Phi_a \hat{x}_k^a + K_{k+1}^a [z_{k+1} - H_a \Phi_a \hat{x}_k^a] \quad (4.2.25)$$

with covariance and gain equations

$$M_{k+1}^a = \Phi_a P_k^a \Phi_a^T + Q_{nb} + \Psi_a Q \Psi_a^T \quad (4.2.26)$$

$$K_{k+1}^a = M_{k+1}^a H_a^T (H_a M_{k+1}^a H_a^T + R)^{-1} \quad (4.2.27)$$

$$P_{k+1}^a = (I - K_{k+1}^a H_a) M_{k+1}^a \quad (4.2.28)$$

Note that in (4.2.25) the measurement z_k produced from (4.2.6) and (4.2.7) using the actual nonGaussian input signal (4.2.9) is used instead of the measurement z_k^a from the augmented system (4.2.21) and (4.2.22).

The MGS filter is initialized with equally-valued weighting terms

$$w_0^j = \frac{1}{3}, \quad \text{for } j = 1, 2, 3 \quad (4.2.29)$$

A Markov transition probability matrix Θ_{bS}^{ij} , consisting of θ_{bS}^{ij} elements, is configured with a high probability that the bias term does not switch from one value to another, and a low probability that the bias term does switch, such as

$$\Theta_{bS}^{ij} = \begin{bmatrix} .95 & .025 & .025 \\ .025 & .95 & .025 \\ .025 & .025 & .95 \end{bmatrix} \quad (4.2.30)$$

The initial values of the state, state estimate, and error covariance for the MGS adaptive filter are

$$x_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \hat{x}_0 = \begin{bmatrix} 0.0004 \\ 0.0135 \end{bmatrix} \quad P_o = \begin{bmatrix} 10000 & 0 \\ 0 & 10000 \end{bmatrix} \quad (4.2.31)$$

and for the augmented Kalman filter are

$$x_0^a = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \hat{x}_0^a = \begin{bmatrix} 0.0004 \\ 0.0135 \\ 0 \\ 0 \end{bmatrix} \quad P_o^a = \begin{bmatrix} 10000 & 0 & 0 & 0 \\ 0 & 10000 & 0 & 0 \\ 0 & 0 & 10000 & 0 \\ 0 & 0 & 0 & 10000 \end{bmatrix} \quad (4.2.32)$$

Figure 9a shows the first state x_{k+1}^1 and the overall state estimate \hat{x}_{k+1}^1 from the MGS adaptive filter, while Figure 9b shows the first state x_{k+1}^1 and the overall state estimate \hat{x}_{k+1}^{a1} from the augmented Kalman filter. Figure 10a and 10b show similar estimation results for the second state x_{k+1}^2 . The MGS adaptive filter performs very well, with its estimates tracking the states almost exactly. The augmented Kalman filter however introduces considerable delay in its estimates. Table 5 shows a normalized mean-square-error percentage measure for the state estimates produced by

$$\text{mse \%} = \frac{E[(x - \hat{x})^2]}{E[x^2]} \times 100 \quad (4.2.33)$$

The mse percentages for the augmented Kalman filter are considerably higher than the mse percentages for the MGS adaptive filter due to the delay in the augmented Kalman filter estimates. Figure 11a shows the weighting terms for each of the three b_i, S_i pairs. Figure 11b compares the three weighting terms with the stochastic FM input signal, showing how the weights switch according to the amplitude of the input signal.

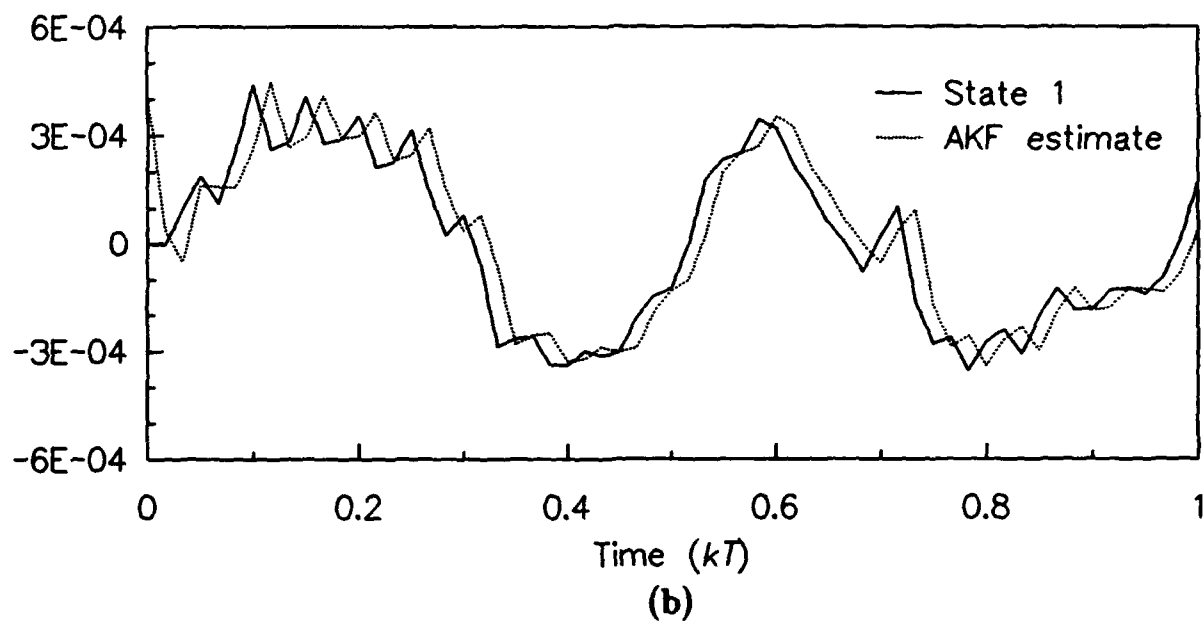
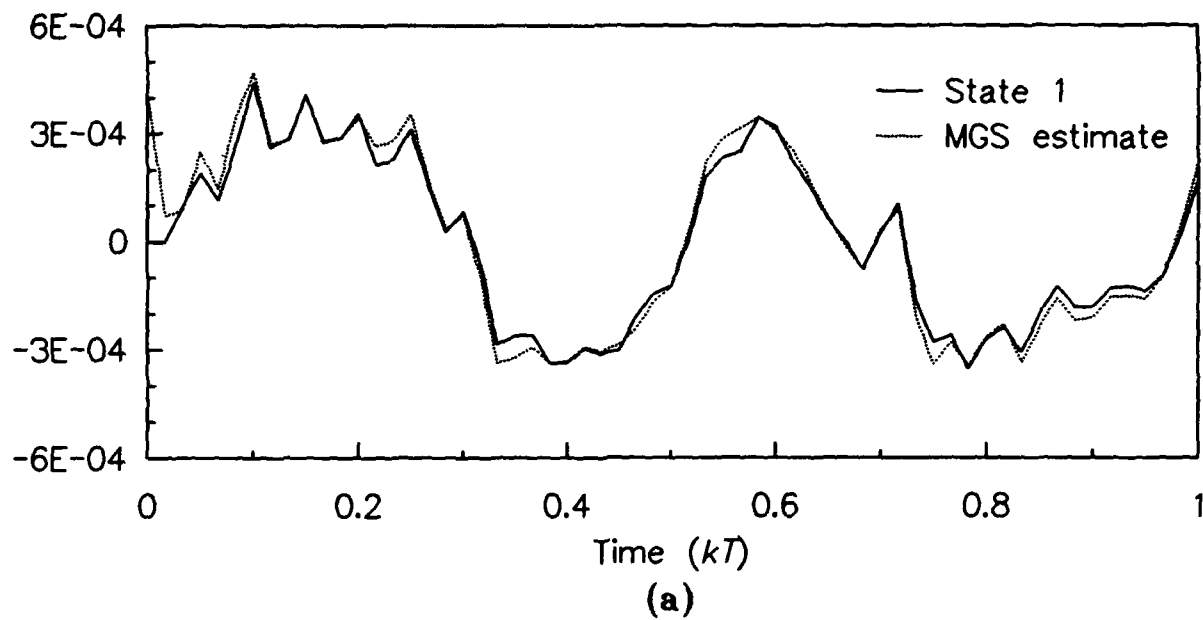


Figure 9. (a) State 1 and MGS adaptive filter estimate,
(b) state 1 and augmented Kalman filter estimate.

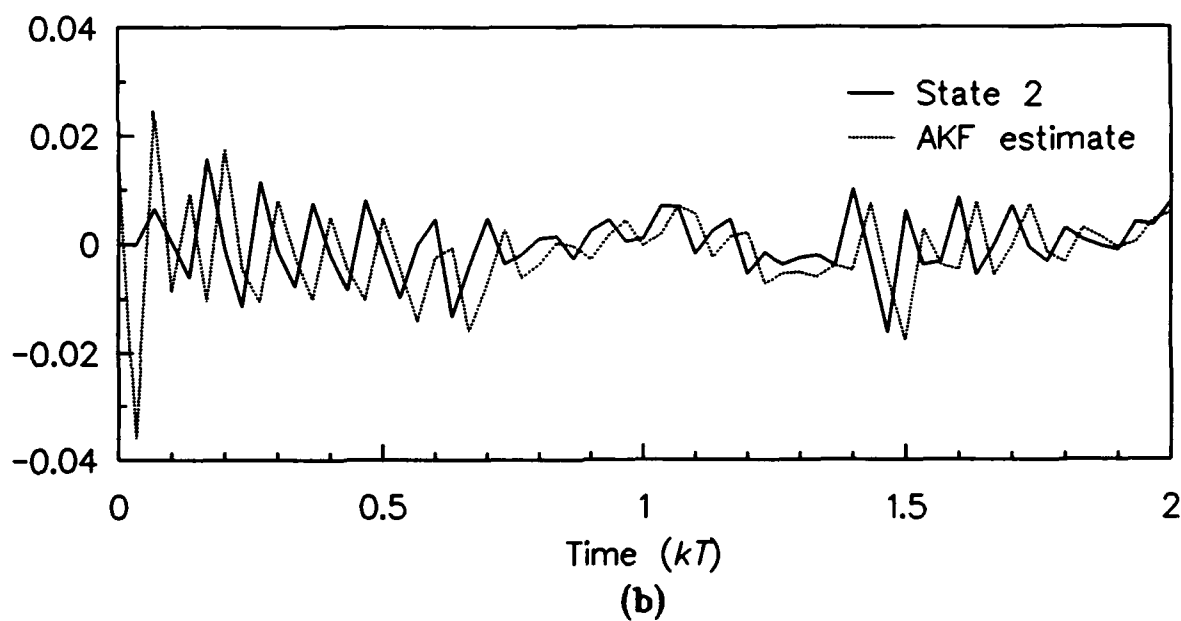
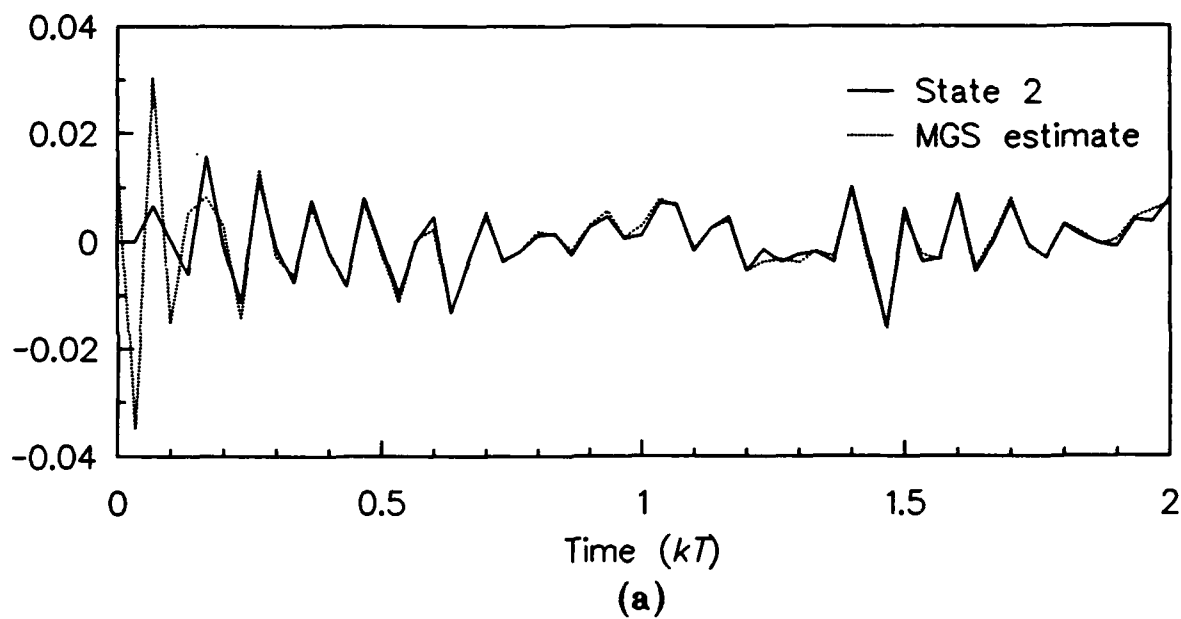


Figure 10. (a) State 2 and MGS adaptive filter estimate,
(b) state 2 and augmented Kalman filter estimate.

Table 5.

Normalized mean-square-error percentage for state estimates.

| State | MGS filter | Augmented filter |
|-------|---------------|---------------------|
| 1 | 1.85 | 14.76 |
| 2 | 3.35 | 227.67 |

$$\text{mse \%} = \frac{E[(x - \hat{x})^2]}{E[x^2]} \times 100$$

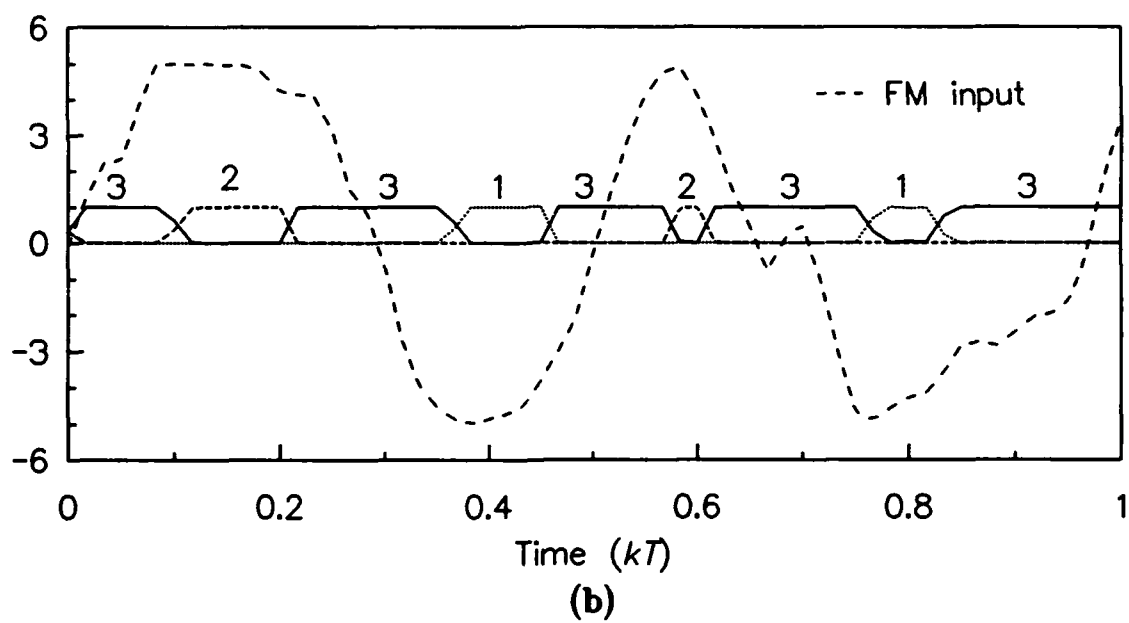
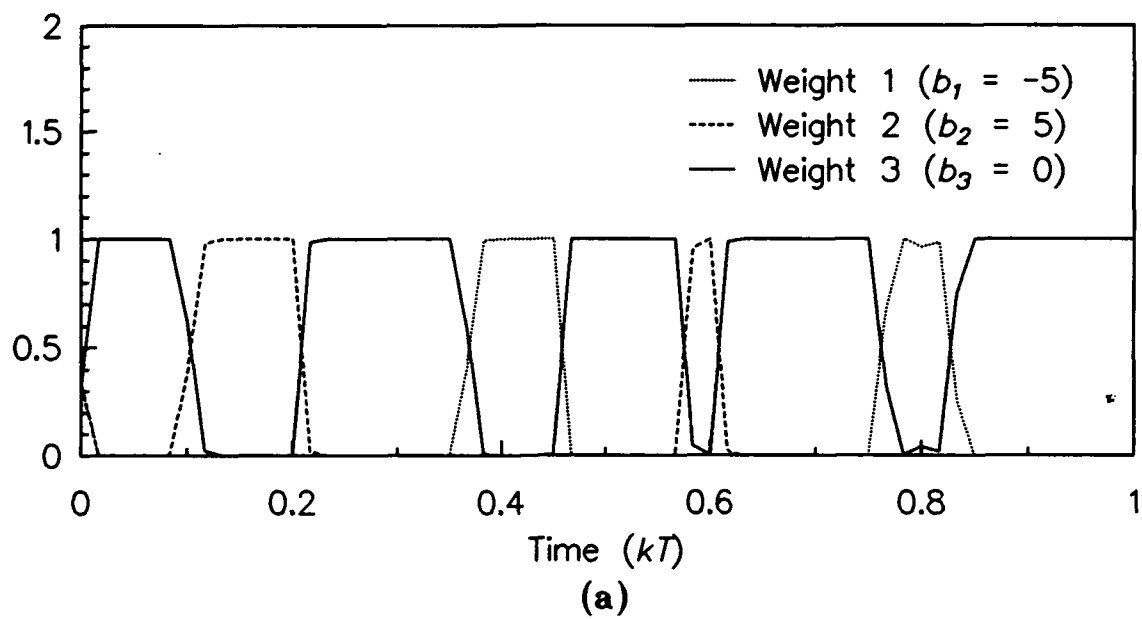


Figure 11. (a) Weighting terms, (b) weighting terms compared with stochastic FM signal.

4.3 Necessary Condition for Effective Estimation

The conditional density (4.1.21), which appears in the weighting term (4.1.27) is of prime importance in determining whether or not the MGS adaptive filter will work properly for a given system. As previously stated, the i^{th} estimator (4.1.16) based on the bias and covariance pair that most closely matches the actual bias and covariance pair of the modeled system will have a corresponding weighting term that tends closer to one, while the weights of the other mismatched estimators will tend towards zero. The bias and covariance pairs influence the weighting terms through the conditional density (4.1.21), rewritten here as the n -dimensional (z_{k+1} is an $n \times 1$ vector) Gaussian conditional density

$$p(z_{k+1} | b_i, S_i, Z_k) = \frac{1}{(2\pi)^{n/2} |R_{k+1}^i|^{1/2}} e^{-\frac{1}{2} r_{k+1}^{iT} R_{k+1}^{i-1} r_{k+1}^i} \quad (4.3.1)$$

where

$$R_{k+1}^i = H M_{k+1}^i H^T + D S_i D^T + R \quad (4.3.2)$$

is defined as the i^{th} measurement covariance from Kalman gain (4.1.18) and

$$r_{k+1}^i = z_{k+1} - D b_i - H(\Phi \hat{x}_k^i + \Gamma b_i) \quad (4.3.3)$$

is defined as the i^{th} measurement residual from Kalman estimator (4.1.16). Essentially, the measurement residual of the matched filter (with the bias and covariance pair that most closely matches the actual pair of the modeled system) will have the smallest expected value, typically zero mean. The residuals of all the

other mismatched filters will be biased. Under the Gaussian assumption, the probability of the matched filter will be the largest among all the filters.

In the previous simulation examples of chapters 3 and 4, this method of calculating the conditional measurement probabilities (4.3.1) and i^{th} estimator weighting terms (4.1.26) works very well, producing very accurate state estimates. However, in the case of the force to acceleration modal system described in [1] and by (A.5) of appendix A, this method did not perform well. Upon analysis of several simulations, it appeared that the statistical steady-state value (that is, the steady-state value of the expected value) of the measurement residual of *each* i^{th} filter converged to zero. Only the correctly matched filter is supposed to have a statistical steady-state (SSS) value of zero, while the residuals of the mismatched filters are supposed to be biased. Yet, in this case, *none* of the residuals were biased. Therefore, the MGS algorithm could not detect which filter possessed the correct bias and covariance pair, and the weights of all the filters converged to the same value.

Upon investigation, it has been determined that the SSS value of the i^{th} measurement residual (4.3.3) is not only a function of the bias term b_i , but is also a function of the dc gain of the system transfer function from u_k to z_k

$$G(z)|_{z=1} = G_{dc} = H(I - \Phi)^{-1}\Gamma + D \quad (4.3.4)$$

The dc gains of the systems used in the previous simulations of chapters 3 and 4 are nonzero, producing different SSS residual values due to the different bias and covariance pairs. As shown in appendix A, the dc gain of the force to acceleration modal system is zero, producing equal SSS residual values despite differences in the bias and covariance pairs.

In order to show the dependency of the SSS residual value upon the system dc gain, first rearrange the i^{th} Kalman filter equation (4.1.16) as

$$\hat{x}_{k+1}^i = (I - K_{k+1}^i H)(\Phi \hat{x}_k^i + \Gamma b_i) + K_{k+1}^i (z_{k+1} - D b_i) \quad (4.3.5)$$

and take the expected value

$$E[\hat{x}_{k+1}^i] = (I - K_{k+1}^i H)(\Phi E[\hat{x}_k^i] + \Gamma b_i) + K_{k+1}^i (E[z_{k+1}] - D b_i) \quad (4.3.6)$$

Equation (4.3.6) requires the expected value of the measurement model (4.1.2) at iteration $k+1$, given as

$$E[z_{k+1}] = H E[x_{k+1}] + D u_{k+1} + E[v_{k+1}] \quad (4.3.7)$$

with u_{k+1} being the actual input signal, assumed known for this development. Substituting the plant model (4.1.1) for x_{k+1} in (4.3.7) gives

$$\begin{aligned} E[z_{k+1}] &= H E[\Phi x_k + \Gamma u_k + \Psi w_k] + D u_{k+1} + E[v_{k+1}] \\ &= H \Phi E[x_k] + H \Gamma u_k + H \Psi E[w_k] + D u_{k+1} + E[v_{k+1}] \\ &= H \Phi E[x_k] + H \Gamma u_k + D u_{k+1} \end{aligned} \quad (4.3.8)$$

since w_k and v_{k+1} are each zero mean. Substituting (4.3.8) into (4.3.6) produces

$$E[\hat{x}_{k+1}^i] = (I - K_{k+1}^i H)(\Phi E[\hat{x}_k^i] + \Gamma b_i) + K_{k+1}^i (H \Phi E[x_k] + H \Gamma u_k + D u_{k+1} - D b_i) \quad (4.3.9)$$

Now, let (4.3.9) reach steady-state with $E[\hat{x}_{k+1}^i] = E[\hat{x}_k^i] = \hat{x}_{ss}^i$ and $u_{k+1} = u_k = u$.

The SSS value of (4.3.9) is then

$$\hat{x}_{ss}^i = (I - K_{ss}^i H)(\Phi \hat{x}_{ss}^i + \Gamma b_i) + K_{ss}^i (H \Phi x_{ss} + H \Gamma u + Du - Db_i) \quad (4.3.10)$$

To find x_{ss} , the SSS value of the state model (4.1.1), first find the expected value

$$\begin{aligned} E[x_{k+1}] &= \Phi E[x_k] + \Gamma u_k + \Psi E[w_k] \\ &= \Phi E[x_k] + \Gamma u_k \end{aligned}$$

and let $E[x_{k+1}] = E[x_k] = x_{ss}$ and $u_k = u$ at steady-state

$$x_{ss} = \Phi x_{ss} + \Gamma u$$

Solving for x_{ss} gives

$$x_{ss} = (I - \Phi)^{-1} \Gamma u \quad (4.3.11)$$

and substituting (4.3.11) into (4.3.10) produces

$$\hat{x}_{ss}^i = (I - K_{ss}^i H)(\Phi \hat{x}_{ss}^i + \Gamma b_i) + K_{ss}^i H \Phi (I - \Phi)^{-1} \Gamma u + K_{ss}^i (H \Gamma u + Du - Db_i) \quad (4.3.12)$$

Now, the $\Phi(I - \Phi)^{-1}$ term of (4.3.12) can be rewritten as the identity

$$\Phi(I - \Phi)^{-1} = (I - \Phi)^{-1} - I \quad (4.3.13)$$

since post-multiplying both sides of (4.3.13) by $(I - \Phi)$ shows the equality

$$\Phi(I - \Phi)^{-1}(I - \Phi) = (I - \Phi)^{-1}(I - \Phi) - (I - \Phi)$$

$$\Phi = I - I + \Phi$$

$$\Phi = \Phi$$

Substituting (4.3.13) into (4.3.12) and expanding gives

$$\begin{aligned}\hat{x}_{ss}^i &= (I - K_{ss}^i H) \Phi \hat{x}_{ss}^i + (I - K_{ss}^i H) \Gamma b_i + K_{ss}^i H (I - \Phi)^{-1} \Gamma u \\ &\quad - K_{ss}^i H \Gamma u + K_{ss}^i H \Gamma u + K_{ss}^i D u - K_{ss}^i D b_i\end{aligned}$$

$$\begin{aligned}\hat{x}_{ss}^i &= (I - K_{ss}^i H) \Phi \hat{x}_{ss}^i + (I - K_{ss}^i H) \Gamma b_i + K_{ss}^i H (I - \Phi)^{-1} \Gamma u \\ &\quad + K_{ss}^i D u - K_{ss}^i D b_i\end{aligned}$$

$$\hat{x}_{ss}^i = (I - K_{ss}^i H) \Phi \hat{x}_{ss}^i + (I - K_{ss}^i H) \Gamma b_i - K_{ss}^i D b_i + K_{ss}^i [H(I - \Phi)^{-1} \Gamma + D] u$$

$$[I - (I - K_{ss}^i H) \Phi] \hat{x}_{ss}^i = (\Gamma - K_{ss}^i H \Gamma - K_{ss}^i D) b_i + K_{ss}^i [H(I - \Phi)^{-1} \Gamma + D] u$$

or

$$\hat{x}_{ss}^i = [I - (I - K_{ss}^i H) \Phi]^{-1} \{[\Gamma - K_{ss}^i (H \Gamma + D)] b_i + K_{ss}^i [H(I - \Phi)^{-1} \Gamma + D] u\} \quad (4.3.14)$$

With x_{ss} (4.3.11) and \hat{x}_{ss}^i (4.3.14) known, the SSS value of the i^{th} measurement residual (4.3.3) can now be found. Taking the expected value of (4.3.3) and using (4.3.8) produces

$$E[r_{k+1}^i] = H \Phi E[x_k] + H \Gamma u_k + D u_{k+1} - D b_i - H \Phi E[\hat{x}_k^i] - H \Gamma b_i \quad (4.3.15)$$

Letting (4.3.15) reach steady-state, with $E[r_{k+1}^i] = r_{ss}^i$ and $u_{k+1} = u_k = u$, the result is

$$r_{ss}^i = H\Phi x_{ss} + H\Gamma u + Du - Db_i - H\Phi \hat{x}_{ss}^i - H\Gamma b_i \quad (4.3.16)$$

Using (4.3.11) and (4.3.13), $H\Phi x_{ss}$ is rewritten as

$$H\Phi x_{ss} = H\Phi(I - \Phi)^{-1}\Gamma u = H(I - \Phi)^{-1}\Gamma u - H\Gamma u \quad (4.3.17)$$

Substituting (4.3.17) into (4.3.16) produces a cancellation of terms

$$\begin{aligned} r_{ss}^i &= H(I - \Phi)^{-1}\Gamma u - H\Gamma u + H\Gamma u + Du - Db_i - H\Phi \hat{x}_{ss}^i - H\Gamma b_i \\ &= H(I - \Phi)^{-1}\Gamma u + Du - Db_i - H\Phi \hat{x}_{ss}^i - H\Gamma b_i \\ &= [H(I - \Phi)^{-1}\Gamma + D]u - (H\Gamma + D)b_i - H\Phi \hat{x}_{ss}^i \end{aligned} \quad (4.3.18)$$

Using (4.3.14) in (4.3.18) gives

$$\begin{aligned} r_{ss}^i &= [H(I - \Phi)^{-1}\Gamma + D]u - (H\Gamma + D)b_i \\ &\quad - H\Phi[I - (I - K_{ss}^i H)\Phi]^{-1}\{[\Gamma - K_{ss}^i(H\Gamma + D)]b_i + K_{ss}^i[H(I - \Phi)^{-1}\Gamma + D]u\} \end{aligned} \quad (4.3.19)$$

The dc gain G_{dc} (4.3.4) appears in two places in (4.3.19) rewritten as

$$r_{ss}^i = G_{dc}u - (H\Gamma + D)b_i - H\Phi[I - (I - K_{ss}^i H)\Phi]^{-1}\{[\Gamma - K_{ss}^i(H\Gamma + D)]b_i + K_{ss}^i G_{dc}u\} \quad (4.3.20)$$

However, (4.3.20) can be simplified further using additional algebraic manipulation.

The $H\Gamma + D$ term can be rewritten by combining the dc gain (4.3.4) and the identity (4.3.13) as

$$\begin{aligned} G_{dc} &= H[\Phi(I - \Phi)^{-1} + I]\Gamma + D \\ &= H\Phi(I - \Phi)^{-1}\Gamma + H\Gamma + D \end{aligned}$$

so that

$$H\Gamma + D = G_{dc} - H\Phi(I - \Phi)^{-1}\Gamma \quad (4.3.21)$$

Substituting (4.3.21) into (4.3.20) and rearranging terms gives

$$\begin{aligned} r_{ss}^i &= G_{dc}u - (G_{dc} - H\Phi(I - \Phi)^{-1}\Gamma)b_i \\ &\quad - H\Phi[I - (I - K_{ss}^i H)\Phi]^{-1}\{[\Gamma - K_{ss}^i(G_{dc} - H\Phi(I - \Phi)^{-1}\Gamma)]b_i + K_{ss}^i G_{dc}u\} \\ r_{ss}^i &= G_{dc}u - G_{dc}b_i + H\Phi(I - \Phi)^{-1}\Gamma b_i \\ &\quad - H\Phi[I - (I - K_{ss}^i H)\Phi]^{-1}\{\Gamma b_i - K_{ss}^i G_{dc}b_i + K_{ss}^i H\Phi(I - \Phi)^{-1}\Gamma b_i + K_{ss}^i G_{dc}u\} \\ r_{ss}^i &= G_{dc}(u - b_i) + H\Phi(I - \Phi)^{-1}\Gamma b_i \\ &\quad - H\Phi[I - (I - K_{ss}^i H)\Phi]^{-1}\{[I + K_{ss}^i H\Phi(I - \Phi)^{-1}]\Gamma b_i + K_{ss}^i G_{dc}(u - b_i)\} \end{aligned} \quad (4.3.22)$$

Next, the $[I - (I - K_{ss}^i H)\Phi]^{-1}$ term can be rearranged as

$$[I - (I - K_{ss}^i H)\Phi]^{-1} = [(I - \Phi) + K_{ss}^i H\Phi]^{-1} \quad (4.3.23)$$

and factoring out an $(I - \Phi)$ term produces

$$[(I - \Phi) + K_{..}^i H \Phi]^{-1} = [(I + K_{..}^i H \Phi (I - \Phi)^{-1})(I - \Phi)]^{-1} \quad (4.3.24)$$

Using the identity

$$(AB)^{-1} = B^{-1}A^{-1}$$

if A and B are nonsingular square matrices, setting $A = I + K_{..}^i H \Phi (I - \Phi)^{-1}$ and $B = (I - \Phi)$, (4.3.24) is equal to

$$[(I + K_{..}^i H \Phi (I - \Phi)^{-1})(I - \Phi)]^{-1} = (I - \Phi)^{-1}(I + K_{..}^i H \Phi (I - \Phi)^{-1})^{-1} \quad (4.3.25)$$

Combining (4.3.23) - (4.3.25) therefore results in

$$[I - (I - K_{..}^i H) \Phi]^{-1} = (I - \Phi)^{-1}(I + K_{..}^i H \Phi (I - \Phi)^{-1})^{-1} \quad (4.3.26)$$

Substituting (4.3.26) into (4.3.22) gives

$$\begin{aligned} r_{..}^i &= G_{dc}(u - b_i) + H \Phi (I - \Phi)^{-1} \Gamma b_i \\ &\quad - H \Phi (I - \Phi)^{-1} (I + K_{..}^i H \Phi (I - \Phi)^{-1})^{-1} \{ [I + K_{..}^i H \Phi (I - \Phi)^{-1}] \Gamma b_i + K_{..}^i G_{dc}(u - b_i) \} \end{aligned}$$

and multiplying through gives a cancellation of terms

$$\begin{aligned} r_{..}^i &= G_{dc}(u - b_i) + H \Phi (I - \Phi)^{-1} \Gamma b_i - H \Phi (I - \Phi)^{-1} \Gamma b_i \\ &\quad - H \Phi (I - \Phi)^{-1} (I + K_{..}^i H \Phi (I - \Phi)^{-1})^{-1} K_{..}^i G_{dc}(u - b_i) \\ &= G_{dc}(u - b_i) - H \Phi (I - \Phi)^{-1} (I + K_{..}^i H \Phi (I - \Phi)^{-1})^{-1} K_{..}^i G_{dc}(u - b_i) \\ &= [I - H \Phi (I - \Phi)^{-1} (I + K_{..}^i H \Phi (I - \Phi)^{-1})^{-1} K_{..}^i] G_{dc}(u - b_i) \end{aligned} \quad (4.3.27)$$

Substituting (4.3.26) into (4.3.27), r_{ss}^i becomes finally

$$r_{ss}^i = [I - H\Phi[I - (I - K_{ss}^i H)\Phi]^{-1} K_{ss}^i] G_{dc}(u - b_i) \quad (4.3.28)$$

The i^{th} SSS residual value (4.3.28) is clearly a function of the dc gain G_{dc} as well as the bias term b_i . It also is a function of the covariance term S_i through the steady state Kalman gain K_{ss}^i . As a result, a nonzero dc system gain becomes a necessary condition for effective estimation.

If G_{dc} is nonzero, then r_{ss}^i will be different for each bias and covariance pair. The filter with the matched pair that causes r_{ss}^i to be the smallest will also cause r_{k+1}^i (4.3.3) to have the smallest expected value of all the filters. The probability (4.3.1) of this matched filter will then be the largest among all the filters, and will produce the largest weighting term (4.1.26). If G_{dc} is zero, then r_{ss}^i is also zero for all the filters. The differences between the bias and covariance pairs are masked, causing the probabilities (4.3.1), and therefore the weighting terms (4.1.26), of all the filters to converge to the same value. At this point, the MGS algorithm becomes an ineffective estimation scheme. However, the alternate estimation procedures described in the next section may provide a solution to this problem.

4.4 Alternate Estimation Procedures

A possible solution to the problem of zero dc system gain is to determine if an alternate measurement provides a nonzero dc gain. For example, if in a particular zero dc gain system acceleration measurements are taken, changing to velocity measurements may provide a nonzero dc gain. If the actual sensors producing

these measurements cannot be changed, integrating the acceleration measurement data to produce *approximate*-velocity measurement data may allow the MGS algorithm to work effectively.

As shown in [1] and by (A.5) of appendix A, the force to acceleration modal system produces acceleration measurement data at sensor grid points and has a zero dc gain. This system is termed the acceleration modal system, or AMS. Changing this AMS to produce velocity measurement data at sensor grid points results in a new transfer function with a nonzero dc gain. This new system is termed the velocity modal system, or VMS. Typically, changing the actual system cannot be done in practice. However, given a large enough signal to noise ratio between the input signal u_k and measurement noise v_k , a good approximation of the velocity measurement data can be generated by integrating the acceleration measurement data from the AMS using a discrete-time integrator. Redesigning the i^{th} measurement covariance (4.3.2) and residual (4.3.3) to be based on the VMS while actually using the integrated acceleration (*approximate*-velocity) measurement data from the AMS produces proper weighting terms (4.1.26) that allow the MGS algorithm to work effectively.

In order to illustrate this alternate estimation procedure, the system and filter equations for the VMS are developed below. Using a zero-order-hold model for the continuous-time force input to velocity output system described by (A.6) of appendix A, the resulting discrete-time system with added noise terms is

$$x_{k+1} = \Phi x_k + \Gamma u_k + \Psi w_k \quad (4.4.1)$$

$$y_k = H_v x_k + v_k^v \quad (4.4.2)$$

$$u_k = b_k + n_k \quad (4.4.3)$$

where

- x_{k+1} is the state vector
- y_k is the velocity measurement vector
- w_k is a zero mean white Gaussian plant noise process with covariance Q
- v_k^v is a zero mean white Gaussian measurement noise process with covariance R_v
- Φ, Γ, Ψ, H_v are the respective constant transition matrices
- u_k is the vector GS signal model of the actual nonGaussian input signal, comprised of semi-Markov bias vector b_k , and zero mean white Gaussian noise n_k with semi-Markov covariance S_k

and the random quantities x_k , w_k , v_k^v , b_k , and n_k are assumed to be mutually statistically independent. The state model (4.4.1) is exactly the same as the model used previously (4.1.1), so that the states of the new VMS are exactly the same as the states of the previous AMS. The Gaussian sum signal model (4.4.3) is also the same as the model used previously (4.1.3). The measurement model (4.4.2) is different than (4.1.2) because y_k is a velocity measurement while z_k is an acceleration measurement. Other differences in (4.4.2) include a new transition matrix H_v and a new measurement noise process v_k^v with new covariance R_v .

Following the same procedures in section 4.1, the MGS algorithm is applied to the system of (4.4.1) - (4.4.3), producing the overall state estimate

$$\hat{x}_{k+1}^v = \sum_{i=1}^M \hat{x}_{k+1}^{vi} p(b_i, S_i | Y_{k+1}) \quad (4.4.4)$$

where Y_{k+1} is the current velocity measurement sequence $\{y_1, y_2, \dots, y_{k+1}\}$, and \hat{x}_{k+1}^{vi}

is given by the Kalman filter equation

$$\hat{x}_{k+1}^{vi} = \Phi \hat{x}_k^{vi} + \Gamma b_i + K_{k+1}^{vi} [y_{k+1} - H_v (\Phi \hat{x}_k^{vi} + \Gamma b_i)] \quad (4.4.5)$$

with covariance and gain equations

$$M_{k+1}^{vi} = \Phi P_k^{vi} \Phi^T + \Gamma S_i \Gamma^T + \Psi Q \Psi^T \quad (4.4.6)$$

$$K_{k+1}^{vi} = M_{k+1}^{vi} H_v^T (H_v M_{k+1}^{vi} H_v^T + R_v)^{-1} \quad (4.4.7)$$

$$P_{k+1}^{vi} = (I - K_{k+1}^{vi} H_v) M_{k+1}^{vi} \quad (4.4.8)$$

and weighting terms

$$w_{k+1}^{vi} = p(b_i, S_i | Y_{k+1}) = C_{k+1}^v p(y_{k+1} | b_i, S_i, Y_k) \sum_{j=1}^M \theta_{bS}^{ij} w_k^{vj} \quad (4.4.9)$$

with

$$p(y_{k+1} | b_i, S_i, Y_k) = N[H_v(\Phi \hat{x}_k^{vi} + \Gamma b_i), H_v M_{k+1}^{vi} H_v^T + R_v] \quad (4.4.10)$$

$$\theta_{bS}^{ij} = p(b_{k+1} = b_i, S_{k+1} = S_i | b_k = b_j, S_k = S_j) \quad (4.4.11)$$

$$w_k^{vj} = p(b_k = b_j, S_k = S_j | Y_k) \quad (4.4.12)$$

where C_{k+1}^v is the scale factor determined at each time interval such that the sum of all the weighting terms (4.4.9) is equal to one.

The conditional density (4.4.10) of the VMS is a function of the i^{th} velocity measurement covariance

$$R_{k+1}^{vi} = H_v M_{k+1}^{vi} H_v^T + R_v \quad (4.4.13)$$

and the i^{th} velocity measurement residual

$$r_{k+1}^{vi} = y_{k+1} - H_v(\Phi \hat{x}_k^{vi} + \Gamma b_i) \quad (4.4.14)$$

which requires velocity measurement data y_{k+1} . This data is not available, since the actual system being simulated is the AMS which produces acceleration measurement data z_{k+1} . In order to generate the proper probabilities (4.4.10) needed by the weighting terms (4.4.9) for effective MGS estimation, *approximate*-velocity measurement data \tilde{y}_{k+1} , generated from the integral of z_{k+1} of the AMS, is used in place of y_{k+1} in (4.4.14) and (4.4.5). This requires a new Kalman filter

$$\hat{x}_{k+1}^{\tilde{v}i} = \Phi \hat{x}_k^{\tilde{v}i} + \Gamma b_i + K_{k+1}^{vi}[\tilde{y}_{k+1} - H_v(\Phi \hat{x}_k^{\tilde{v}i} + \Gamma b_i)] \quad (4.4.15)$$

and new weighting terms

$$w_{k+1}^{\tilde{v}i} = p(b_i, S_i | \tilde{Y}_{k+1}) = C_{k+1}^{\tilde{v}} p(\tilde{y}_{k+1} | b_i, S_i, \tilde{Y}_k) \sum_{j=1}^M \theta_{bS}^{ij} w_k^{\tilde{v}j} \quad (4.4.16)$$

requiring

$$p(\tilde{y}_{k+1} | b_i, S_i, \tilde{Y}_k) = N[H_v(\Phi \hat{x}_k^{\tilde{v}i} + \Gamma b_i), H_v M_{k+1}^{vi} H_v^T + R_v] \quad (4.4.17)$$

$$w_k^{\tilde{v}j} = p(b_k = b_j, S_k = S_j | \tilde{Y}_k) \quad (4.4.18)$$

The Kalman gain and covariance equations (4.4.6) - (4.4.8) and Markov probabilities (4.4.11) can be used without modification since they are not dependent on the *approximate*-velocity measurement data \tilde{y}_{k+1} .

The overall estimate can be calculated by three different methods. The first

method produces

$$\hat{x}_{k+1}^{\tilde{a}} = \sum_{i=1}^M \hat{x}_{k+1}^i p(b_i, S_i | \tilde{Y}_{k+1}) \quad (4.4.19)$$

where \hat{x}_{k+1}^i is the i^{th} state estimate (4.1.16) from the AMS. There was never a problem in calculating these AMS state estimates. It was the problem in calculating the weighting terms that necessitated an alternate estimation scheme. However, (4.4.19) requires two *separate* parallel banks of Kalman filters: one using (4.1.16) to calculate \hat{x}_{k+1}^i , and a second using (4.4.15) to calculate $w_{k+1}^{\tilde{v}i} = p(b_i, S_i | \tilde{Y}_{k+1})$.

The second method produces

$$\hat{x}_{k+1}^{\tilde{v}} = \sum_{i=1}^M \hat{x}_{k+1}^{\tilde{v}i} p(b_i, S_i | \tilde{Y}_{k+1}) \quad (4.4.20)$$

where $\hat{x}_{k+1}^{\tilde{v}i}$ is the i^{th} state estimate (4.4.15) using *approximate*-velocity measurement data with the VMS. This overall state estimate (4.4.20) requires only one bank of parallel Kalman filters using (4.4.15) to calculate both $\hat{x}_{k+1}^{\tilde{v}i}$ and $w_{k+1}^{\tilde{v}i} = p(b_i, S_i | \tilde{Y}_{k+1})$. This provides good computational savings over calculating (4.4.19). However, (4.4.19) uses the actual AMS measurement data z_k to calculate \hat{x}_{k+1}^i , while (4.4.20) never uses z_k in its direct form.

The third method uses the same state equation (4.4.1) and GS signal model (4.4.3) as the AMS and VMS, but requires the use of a second new measurement formed by combining z_k (4.1.2) and y_k (4.4.2) into

$$\zeta_k = \begin{bmatrix} z_k \\ y_k \end{bmatrix} = H_{\zeta} x_k + D_{\zeta} u_k + v_{\zeta}^k \quad (4.4.21)$$

where

$$H_\zeta = \begin{bmatrix} H \\ H_v \end{bmatrix} \quad D_\zeta = \begin{bmatrix} D \\ 0 \end{bmatrix} \quad v_\zeta = \begin{bmatrix} v_k \\ v_k^v \end{bmatrix} \quad (4.4.22)$$

Again, velocity measurement data y_k is not available, so *approximate*-velocity measurement data \tilde{y}_k is used instead producing

$$\tilde{\zeta}_k = \begin{bmatrix} z_k \\ \tilde{y}_k \end{bmatrix} \quad (4.4.23)$$

The resulting Kalman filter equation is

$$\hat{x}_{k+1}^i = \Phi \hat{x}_k^i + \Gamma b_i + K_{k+1}^i [\tilde{\zeta}_{k+1} - D_\zeta b_i - H_\zeta (\Phi \hat{x}_k^i + \Gamma b_i)] \quad (4.4.24)$$

with covariance and gain equations (independent of \tilde{y}_k)

$$M_{k+1}^i = \Phi P_k^i \Phi^T + \Gamma S_i \Gamma^T + \Psi Q \Psi^T \quad (4.4.25)$$

$$K_{k+1}^i = M_{k+1}^i H_\zeta^T (H_\zeta M_{k+1}^i H_\zeta^T + D_\zeta S_i D_\zeta^T + R_\zeta)^{-1} \quad (4.4.26)$$

$$P_{k+1}^i = (I - K_{k+1}^i H_\zeta) M_{k+1}^i \quad (4.4.27)$$

where the covariance of v_ζ^i is given by

$$R_\zeta = \begin{bmatrix} R & 0 \\ 0 & R_v \end{bmatrix} \quad (4.4.28)$$

and the resulting weighting terms are

$$w_{k+1}^i = p(b_i, S_i | \tilde{\mathcal{Z}}_{k+1}) = C_{k+1}^i p(\tilde{\zeta}_{k+1} | b_i, S_i, \tilde{\mathcal{Z}}_k) \sum_{j=1}^M \theta_{bS}^{ij} w_k^j \quad (4.4.29)$$

where $\tilde{\mathcal{Z}}_{k+1}$ is the current measurement sequence $\{\tilde{\zeta}_1, \tilde{\zeta}_2, \dots, \tilde{\zeta}_k\}$, θ_{bS}^{ij} is given by (4.4.11), and

$$p(\tilde{\zeta}_{k+1} | b_i, S_i, \tilde{\mathcal{Z}}_k) = N[H_\zeta(\Phi \tilde{x}_k^i + \Gamma b_i) + D_\zeta b_i, H_\zeta M_{k+1}^i H_\zeta^T + D_\zeta S_i D_\zeta^T + R_\zeta] \quad (4.4.30)$$

$$w_{\tilde{\zeta}}^j = p(b_k = b_j, S_k = S_j | \tilde{\mathcal{Z}}_k) \quad (4.4.31)$$

Forming measurement data $\tilde{\zeta}_k$ (4.4.23) allows both the state estimate (4.4.24) and weighting term (4.4.29) to be influenced by both z_k and \tilde{y}_k at the same time, while requiring only one parallel bank of Kalman filters. The overall estimate produced by using this third method is

$$\tilde{x}_{k+1}^{\tilde{\zeta}} = \sum_{i=1}^M \tilde{x}_{k+1}^i p(b_i, S_i | \tilde{\mathcal{Z}}_{k+1}) \quad (4.4.32)$$

These alternate estimation procedures will be compared and evaluated in the next section.

4.5 Comparison of the Alternate Estimation Procedures

The alternate estimation procedures described in the previous section are compared and evaluated using a simulation example. The overall state estimate \tilde{x}_{k+1}^v (4.4.4) based on the VMS using true velocity measurement data y_k will also be calculated as a benchmark for comparison to the three alternate overall state estimates. For convenience, Table 6 provides a summary of the overall state estimates used in the comparison. A force input to acceleration output modal system consisting of three modes serves as the simulation example. The modes are

calculated using (A.7) - (A.14) of appendix A. The parameter values of α_i , $\angle c_i$, x , and f_i used in the three mode, six state model are given in Table 7.

Table 6.
Overall state estimates to be compared.

| | | |
|----|---|---|
| a. | $\hat{x}_{k+1}^v = \sum_{i=1}^M \hat{x}_{k+1}^{vi} p(b_i, S_i Y_{k+1})$ | Benchmark estimate. i^{th} state estimate and weighting term based on VMS using true velocity measurement data. |
| b. | $\hat{x}_{k+1}^{\tilde{a}} = \sum_{i=1}^M \hat{x}_{k+1}^i p(b_i, S_i \tilde{Y}_{k+1})$ | i^{th} state estimate based on AMS. Weighting term based on VMS using <i>approximate</i> -velocity measurement data. |
| c. | $\hat{x}_{k+1}^{\tilde{v}} = \sum_{i=1}^M \hat{x}_{k+1}^{\tilde{vi}} p(b_i, S_i \tilde{Y}_{k+1})$ | i^{th} state estimate and weighting term based on VMS using <i>approximate</i> -velocity measurement data. |
| d. | $\hat{x}_{k+1}^{\tilde{z}} = \sum_{i=1}^M \hat{x}_{k+1}^{\tilde{zi}} p(b_i, S_i \tilde{Z}_{k+1})$ | i^{th} state estimate and weighting term based on combined AMS and VMS using acceleration measurement data and <i>approximate</i> -velocity measurement data. |

Table 7.
Parameter values for the three mode model.

| i | α_i | $\angle c_i$ (rad) | x (m) | f_i (Hz) |
|-----|------------|--------------------|---------|------------|
| 1 | 1.446813 | 5.686 | 17.5 | 95.63 |
| 2 | 4.934421 | 0.705 | 35.0 | 67.47 |
| 3 | 13.721530 | 1.598 | 52.5 | 28.02 |

The system transition matrices are found from appendix A equations (A.5) for the AMS and (A.6) for the VMS. Using a zero-order-hold model, a sampling time of $T = 0.001$ seconds, and the values in Table 7, the discrete-time matrices for the AMS are

$$\Phi = \begin{bmatrix} 0.8250 & 0.0009 & 0 & 0 & 0 & 0 \\ -339.2108 & 0.8223 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.9118 & 0.0010 & 0 & 0 \\ 0 & 0 & -173.5593 & 0.9022 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.9846 & 0.0010 \\ 0 & 0 & 0 & 0 & -30.6059 & 0.9577 \end{bmatrix} \quad (4.5.1)$$

$$\Gamma = \begin{bmatrix} 0.0069 & 0.0033 & -0.0015 \\ -2.7667 & -4.5899 & -4.9021 \\ 0.0014 & -0.0040 & -0.0071 \\ -3.0180 & -2.5293 & -0.5295 \\ -0.0059 & 0.0023 & 0.0038 \\ -0.0314 & 0.9420 & -0.8696 \end{bmatrix} \quad (4.5.2)$$

$$\Psi = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.5.3)$$

$$H = \begin{bmatrix} -2807.66 & 16.1818 & -5625.67 & 6.0323 & -729.65 & -11.5741 \\ -7693.42 & 10.9286 & -5815.17 & -5.2798 & 2099.60 & 2.6883 \\ -10046.6 & 2.0778 & -2530.33 & -13.4374 & -1278.69 & 9.0026 \end{bmatrix} \quad (4.5.4)$$

$$D = \begin{bmatrix} 110.7665 & -45.3759 & -166.8950 \\ -45.3759 & -166.8950 & -41.4010 \\ -166.8950 & -41.4010 & -55.5049 \end{bmatrix} \quad (4.5.5)$$

The H_v transition matrix needed for the VMS is

$$H_v = \begin{bmatrix} 16.2043 & 0.0078 & 6.3411 & 0.0313 & -10.9321 & 0.0234 \\ 10.9903 & 0.0213 & -4.9605 & 0.0324 & 0.8409 & -0.0673 \\ 2.1583 & 0.0278 & -13.2985 & 0.0141 & 10.1277 & 0.0410 \end{bmatrix} \quad (4.5.6)$$

The measurement model of the combined AMS and VMS requires H_c and D_c matrices formed using (4.4.22). The plant noise w_k of the state model is zero-mean white Gaussian with covariance

$$Q = \begin{bmatrix} 1.4142 & 0 & 0 \\ 0 & 0.2828 & 0 \\ 0 & 0 & 0.0344 \end{bmatrix} \quad (4.5.7)$$

and is uncorrelated with AMS measurement noise v_k and VMS measurement noise v_k^v . Both are zero-mean white Gaussian with respective covariances

$$R = \begin{bmatrix} 26.36 & 0 & 0 \\ 0 & 19.77 & 0 \\ 0 & 0 & 20.41 \end{bmatrix} \quad R_v = \begin{bmatrix} 0.0264 & 0 & 0 \\ 0 & 0.0198 & 0 \\ 0 & 0 & 0.0204 \end{bmatrix} \quad (4.5.8)$$

The measurement noise covariance R_c of the combined AMS and VMS is given by (4.4.28).

The nonGaussian input signal is the stochastic FM signal generated by

$$u(t) = A_u \sin(2\pi f_u t + k_u \int_0^t m(\tau) d\tau) \quad (4.5.9)$$

where the amplitude is $A_u = 5$, the carrier frequency is $f_u = 10$ Hz, the modulation index is $k_u = 100$, and $m(t)$ is zero-mean white Gaussian noise with variance 1. Figure 12 shows the FM signal for 0.2 second. The probability density function of (4.5.9) has the same form as (4.2.10).

The nonGaussian input signal (4.5.9) will be modeled in discrete-time using the Gaussian sum (GS) signal model described in section 4.1

$$u_k = b_k + n_k \quad (4.5.10)$$

with GS density approximation (4.2.12). For simplicity, the nonGaussian input signal is applied only to mode 1.

Three filters are used in each MGS adaptive filter of Table 6. Using the values of Table 4, the b_i and S_i parameters for the three filters are

$$b_1 = \begin{bmatrix} -5 \\ 0 \\ 0 \end{bmatrix} \quad b_2 = \begin{bmatrix} 5 \\ 0 \\ 0 \end{bmatrix} \quad b_3 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (4.5.11)$$

$$S_1 = S_2 = \begin{bmatrix} 0.04 & 0 & 0 \\ 0 & 0.04 & 0 \\ 0 & 0 & 0.04 \end{bmatrix} \quad S_3 = \begin{bmatrix} 4.28 & 0 & 0 \\ 0 & 4.28 & 0 \\ 0 & 0 & 4.28 \end{bmatrix} \quad (4.5.12)$$

Three of the MGS adaptive filters of Table 6 require *approximate*-velocity measurements. A discrete-time integrator implementing the trapezoidal rule is used, given by [59]

$$\tilde{y}_{k+1} = \tilde{y}_k + \frac{T}{2}(z_{k+1} + z_k) \quad (4.5.13)$$

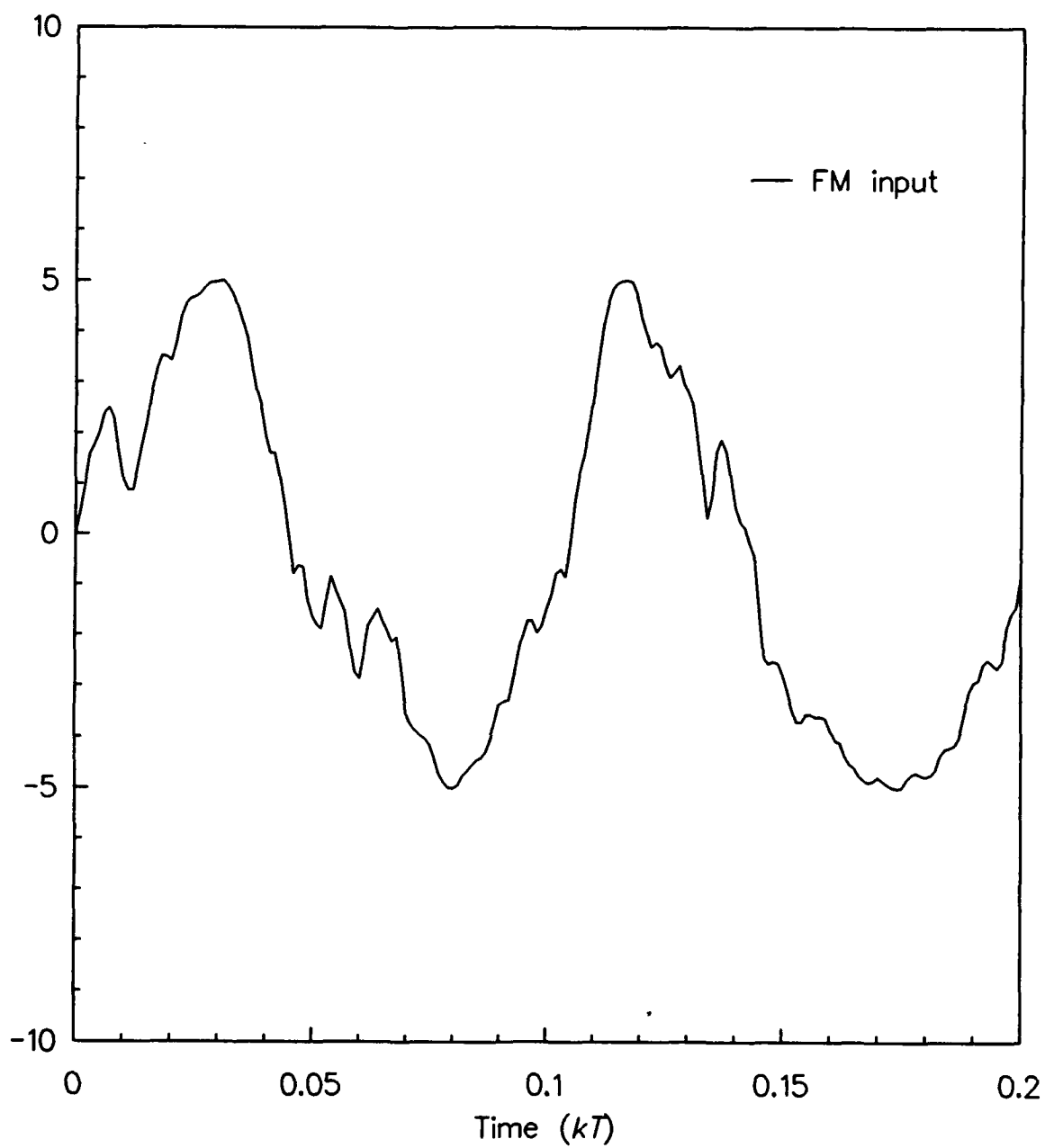


Figure 12. Stochastic FM input signal.

Figure 13a shows the mode 1 acceleration measurement z_k , while Figure 13b shows both the mode 1 actual velocity measurement y_k and *approximate*-velocity measurement \tilde{y}_k . The two velocity measurements track fairly closely under low noise conditions.

Each MGS adaptive filter is initialized with equally-valued weighting terms

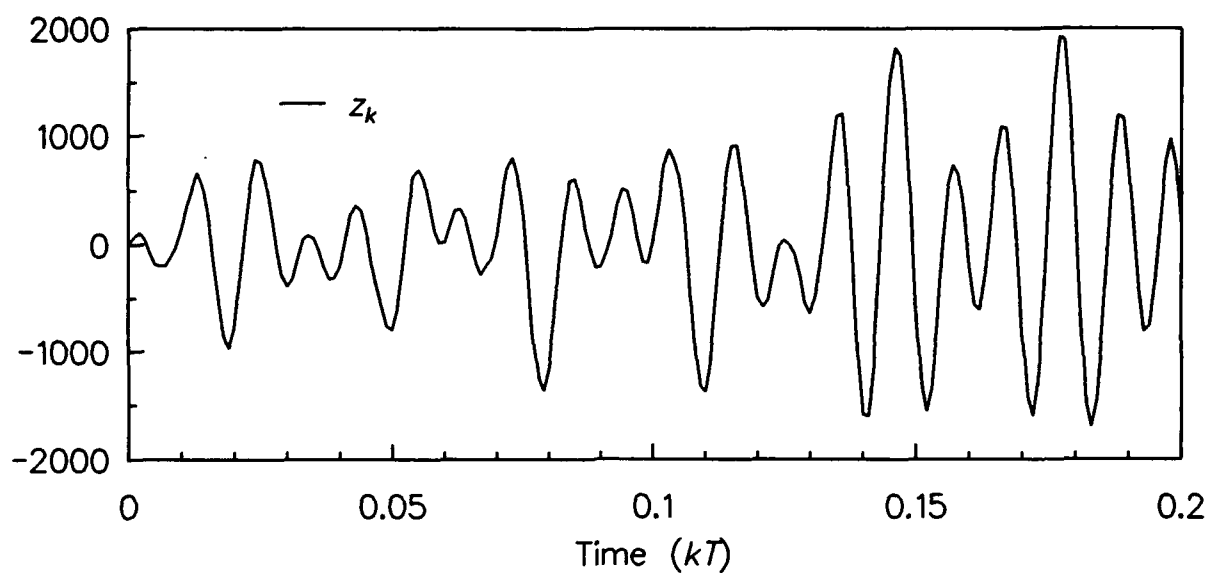
$$w_0^j = \frac{1}{3}, \quad \text{for } j = 1, 2, 3 \quad (4.5.14)$$

A Markov transition probability matrix Θ_{bS}^{ij} , consisting of θ_{bS}^{ij} elements, is configured with a high probability that the bias term does not switch from one value to another, and a low probability that the bias term does switch, such as

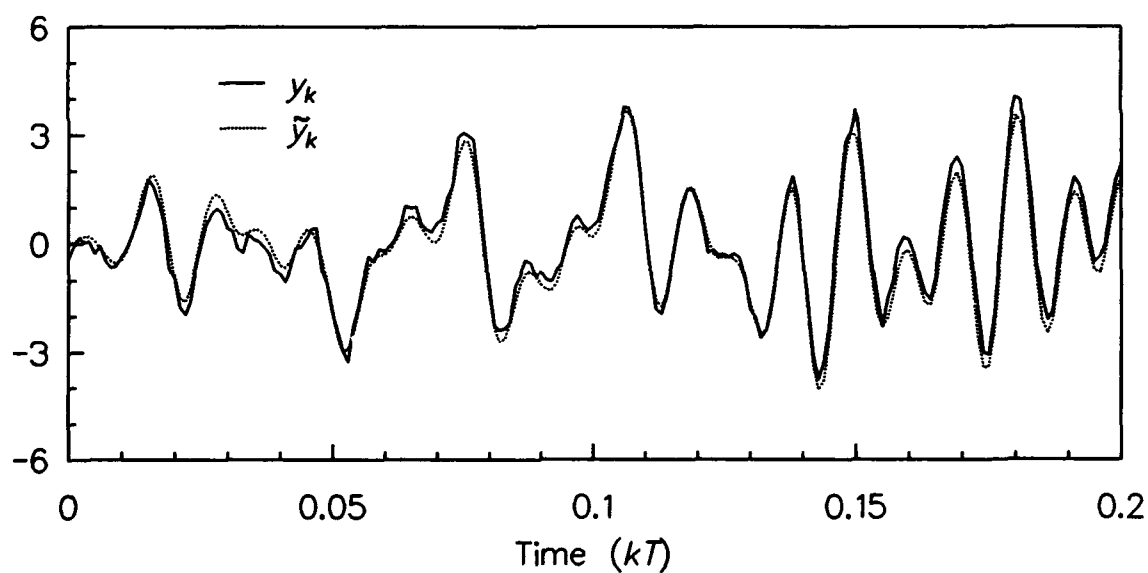
$$\Theta_{bS}^{ij} = \begin{bmatrix} .85 & .075 & .075 \\ .075 & .85 & .075 \\ .075 & .075 & .85 \end{bmatrix} \quad (4.5.15)$$

The initial values of the state, state estimate, and error covariance for each MGS adaptive filter are

$$x_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \hat{x}_0 = \begin{bmatrix} 0.2 \\ 152 \\ 0.2 \\ 62 \\ 0.2 \\ 37 \end{bmatrix} \quad P_o = \begin{bmatrix} 10000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10000 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10000 \end{bmatrix} \quad (4.5.16)$$



(a)



(b)

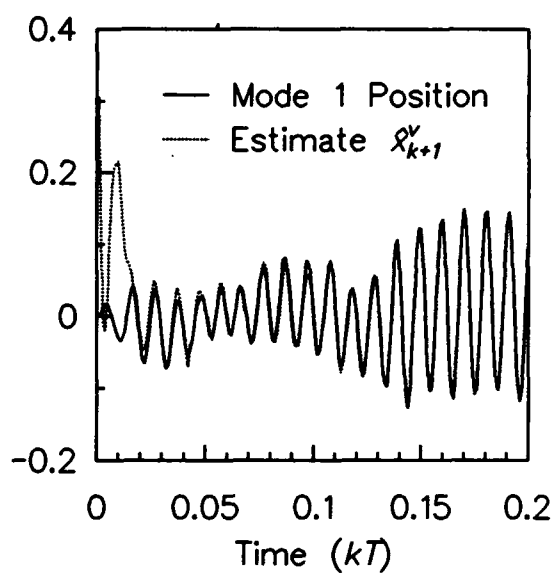
Figure 13. (a) Mode 1 acceleration measurement z_k , (b) mode 1 actual velocity measurement y_k and *approximate*-velocity measurement \tilde{y}_k .

Each plot in Figure 14 shows the position state from mode 1 and an overall position state estimate from a particular MGS adaptive filter. Figures 14a, b, c, and d follow the designations of Table 6: estimator *a*, \hat{x}_{k+1}^v ; estimator *b*, $\hat{\tilde{x}}_{k+1}^a$; estimator *c*, $\hat{\tilde{x}}_{k+1}^v$; estimator *d*, $\hat{\tilde{x}}_{k+1}^a$. Each plot in Figure 15 shows the velocity state from mode 1 and an overall velocity state estimate from a particular MGS adaptive filter. The states and state estimates are shown for mode 2 in Figures 16 and 17, and for 0.4 seconds of mode 3 in Figures 18 and 19. Apart from benchmark estimator *a*, the best estimates appear to come from estimators *d* and *c*, while estimator *b* deviates the most from the actual state.

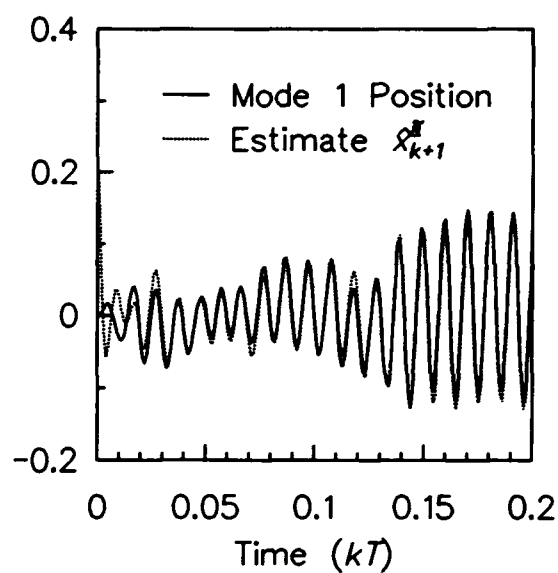
Figure 20a shows the slightly noisy weighting terms used in estimator *a*. Figure 20b shows the weighting terms used in estimators *b* and *c*, which are smoother due to the effect of integrating the acceleration measurements. Figure 20c shows the weighting terms used in estimator *d*, which are also smoother than those of estimator *a*, and reach and remain at their peak values more so than those of estimators *b* and *c*.

Table 8 shows a normalized mean-square-error percentage measure for the state estimators, calculated using (4.2.33). The mse percentages are the lowest for benchmark estimator *a*, followed by estimators *d*, *c*, and *a*.

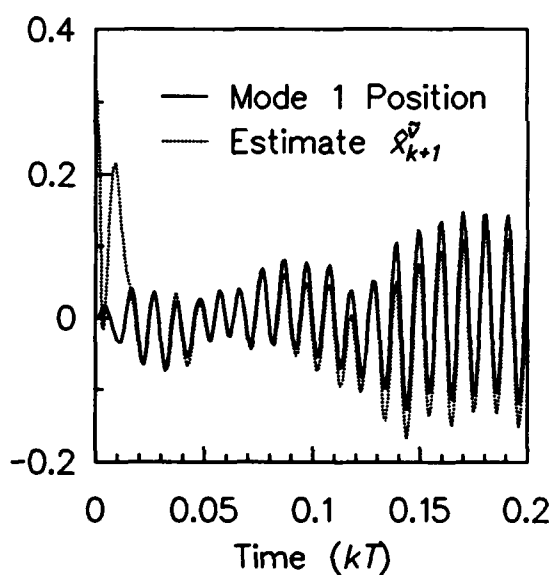
Figure 21a shows the three mode 1 residuals (4.2.3) produced from the MGS adaptive filter using the zero dc gain force to acceleration modal system (A.5) of appendix A. The three residuals become equal once steady-state is reached. Figure 21b shows the three mode 1 residuals produced from the MGS adaptive filter using the nonzero dc gain force to velocity modal system (A.6) of appendix A. The residuals remain biased throughout the simulation.



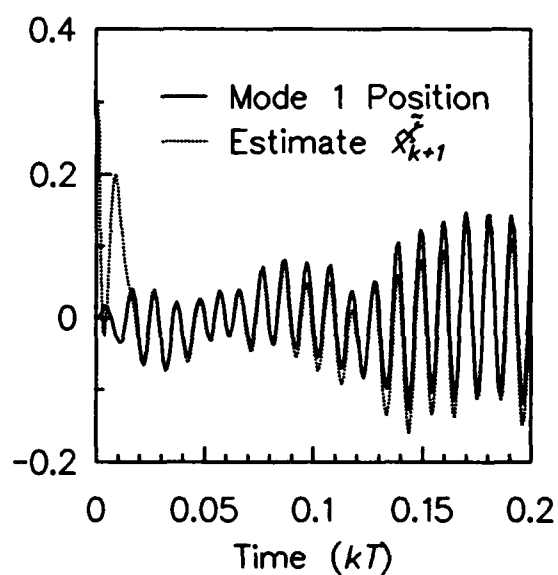
(a)



(b)

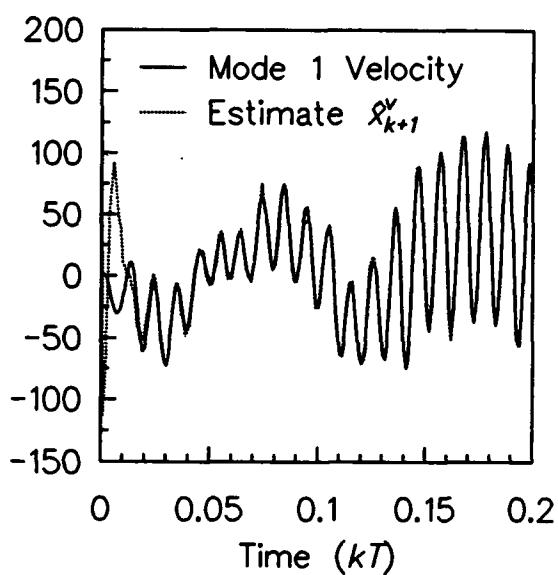


(c)

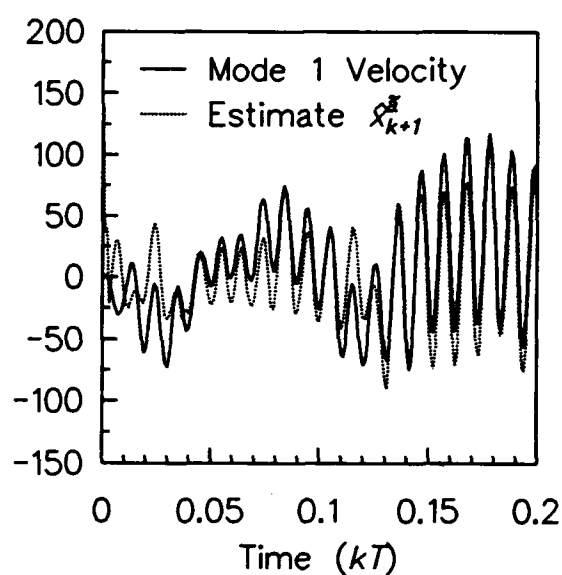


(d)

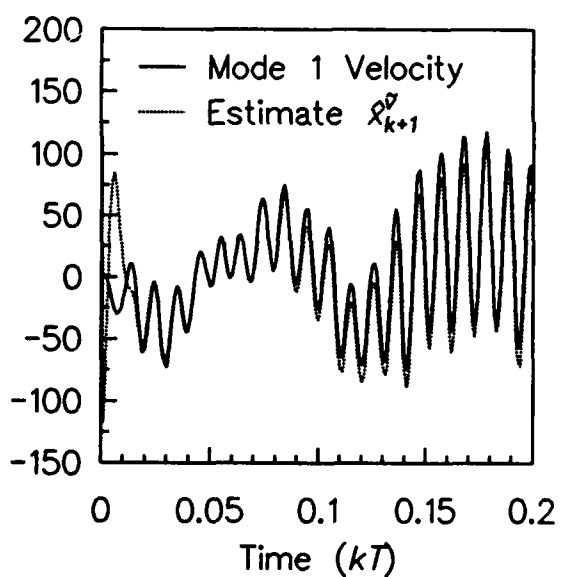
Figure 14. Mode 1 position: (a) State and estimate \hat{x}_{k+1}^v , (b) state and estimate \hat{x}_{k+1}^f , (c) state and estimate $\hat{x}_{k+1}^{\tilde{v}}$, (d) state and estimate $\hat{x}_{k+1}^{\tilde{f}}$.



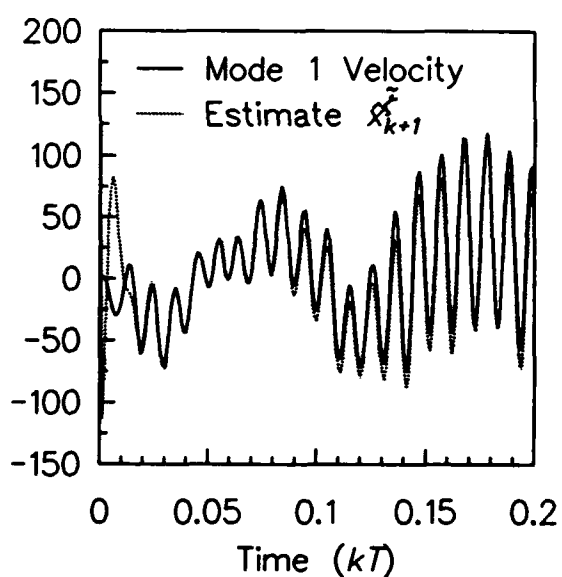
(a)



(b)



(c)



(d)

Figure 15. Mode 1 velocity: (a) State and estimate \hat{x}_{k+1}^v , (b) state and estimate $\hat{x}_{k+1}^{\tilde{a}}$, (c) state and estimate $\hat{x}_{k+1}^{\tilde{v}}$, (d) state and estimate $\hat{x}_{k+1}^{\tilde{\tilde{a}}}$.

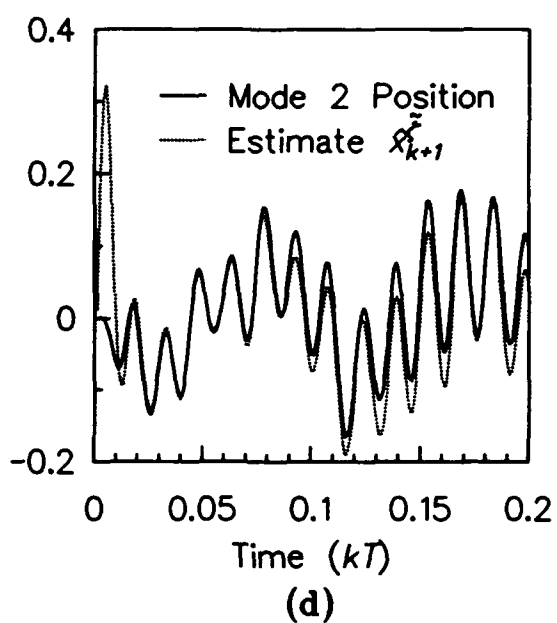
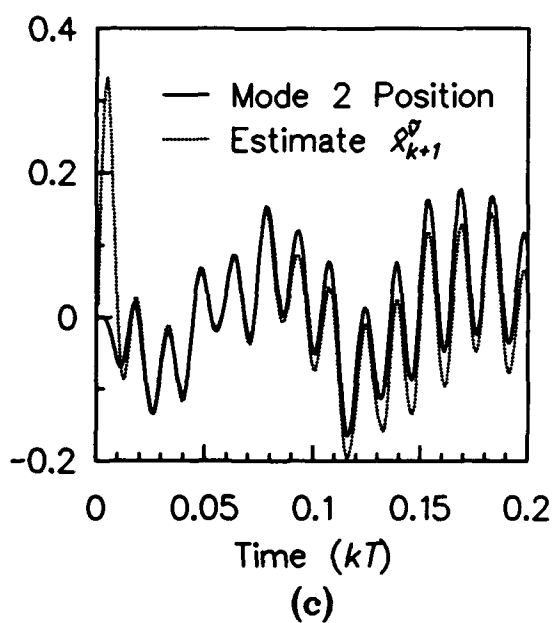
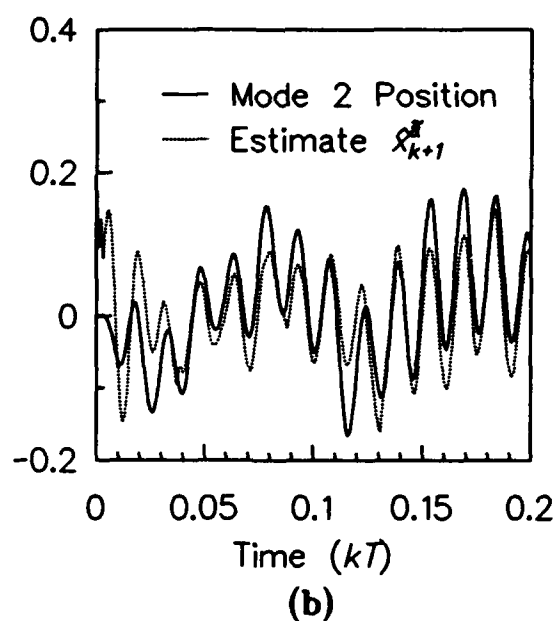
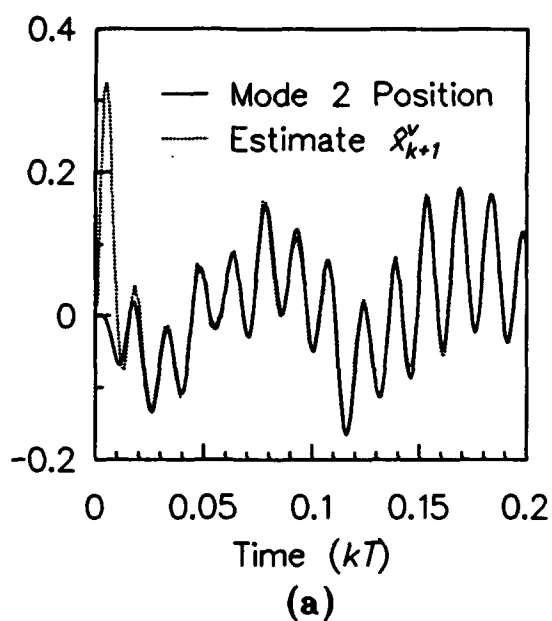


Figure 16. Mode 2 position: (a) State and estimate \hat{x}_{k+1}^v , (b) state and estimate \hat{x}_{k+1}^x , (c) state and estimate $\hat{x}_{k+1}^{\tilde{v}}$, (d) state and estimate $\hat{x}_{k+1}^{\tilde{x}}$.

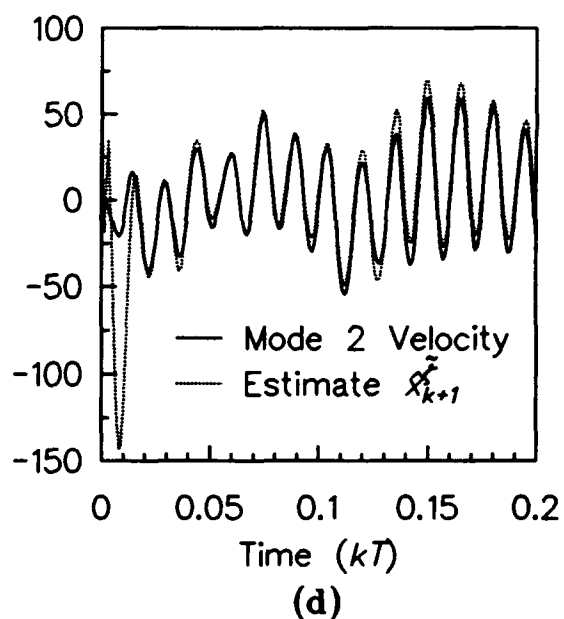
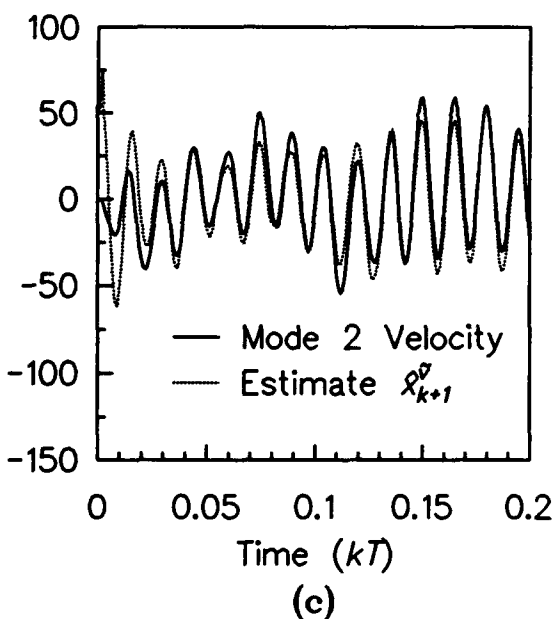
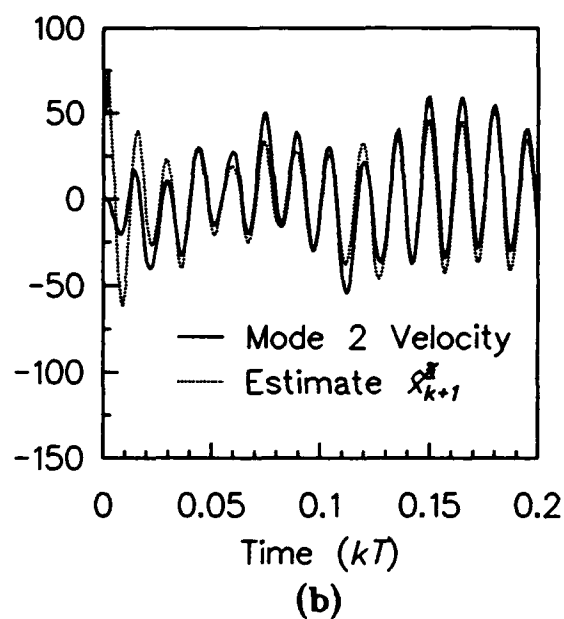
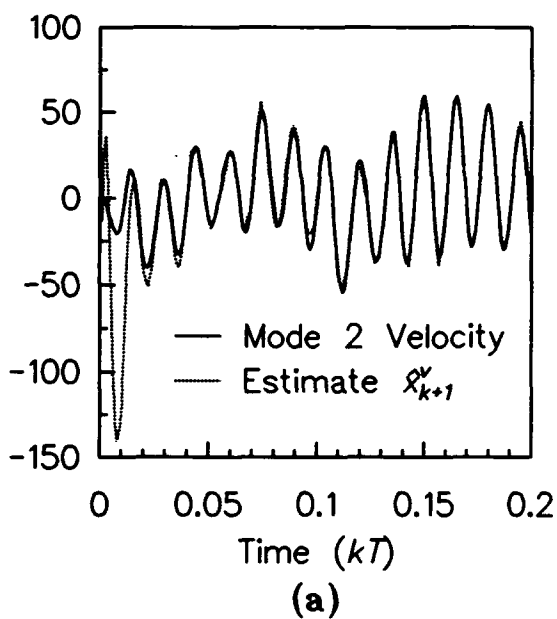
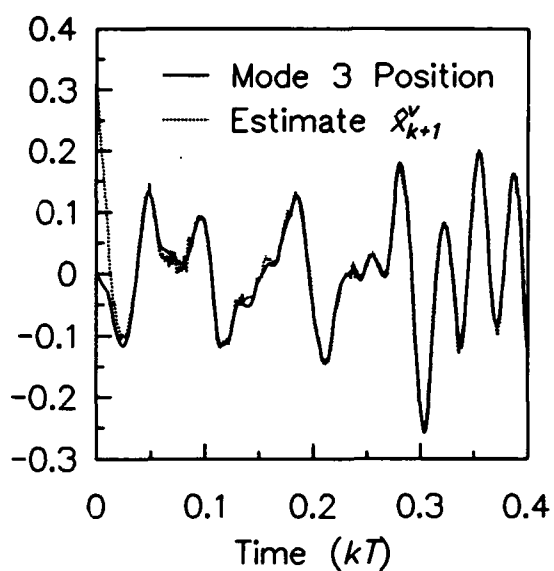
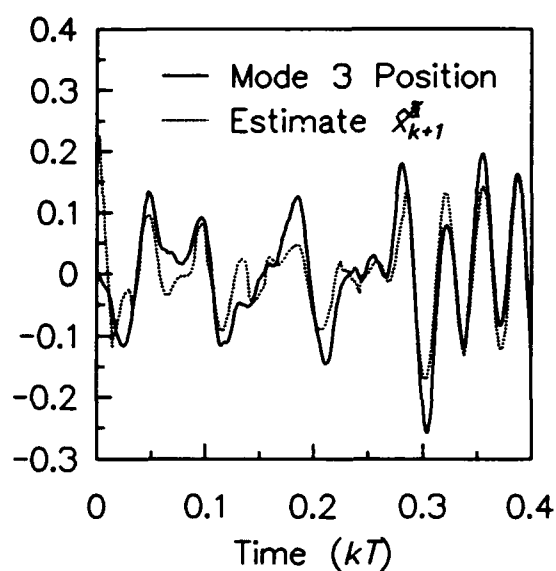


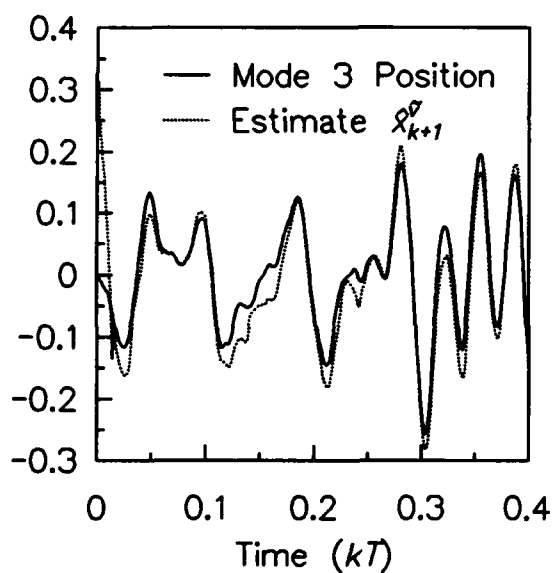
Figure 17. Mode 2 velocity: (a) State and estimate \hat{x}_{k+1}^v , (b) state and estimate $\hat{x}_{k+1}^{\tilde{a}}$, (c) state and estimate $\hat{x}_{k+1}^{\tilde{v}}$, (d) state and estimate $\hat{x}_{k+1}^{\tilde{f}}$.



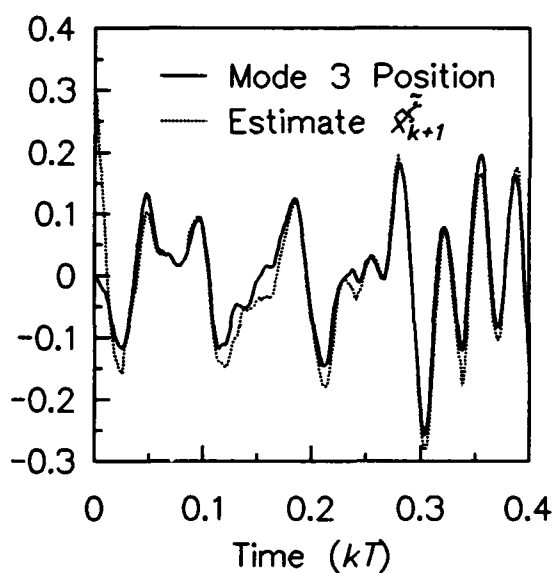
(a)



(b)

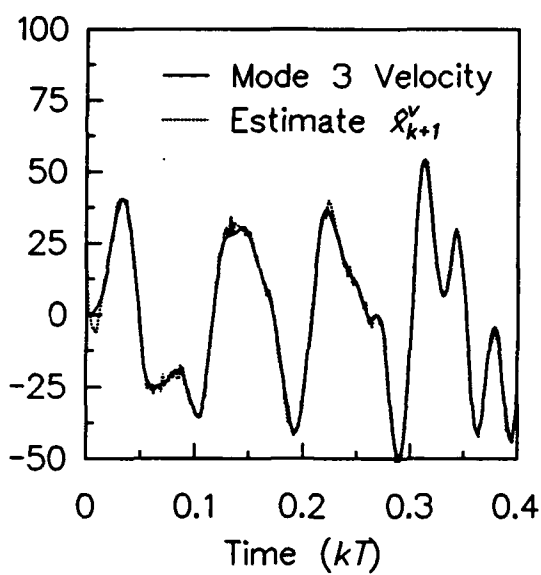


(c)

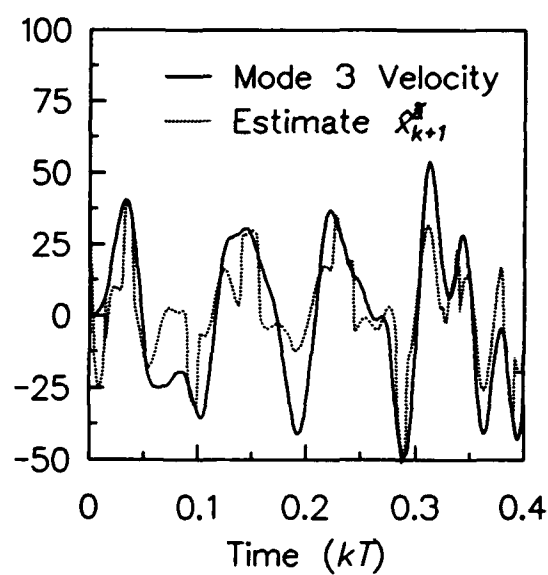


(d)

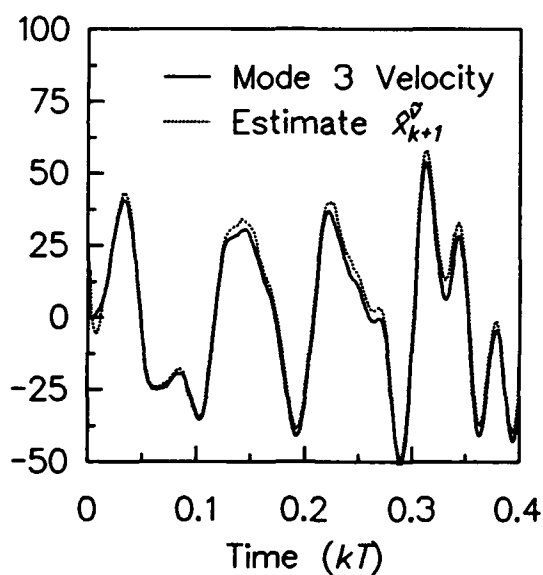
Figure 18. Mode 3 position: (a) State and estimate \hat{x}_{k+1}^v , (b) state and estimate $\hat{x}_{k+1}^{\tilde{x}}$, (c) state and estimate $\hat{x}_{k+1}^{\tilde{v}}$, (d) state and estimate $\hat{x}_{k+1}^{\tilde{\tilde{x}}}$.



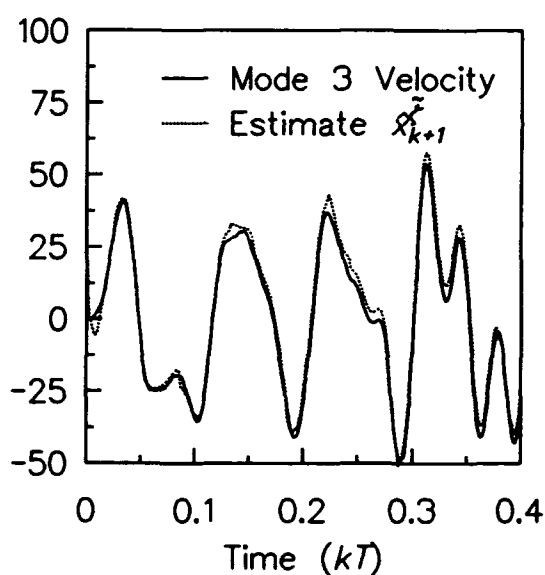
(a)



(b)

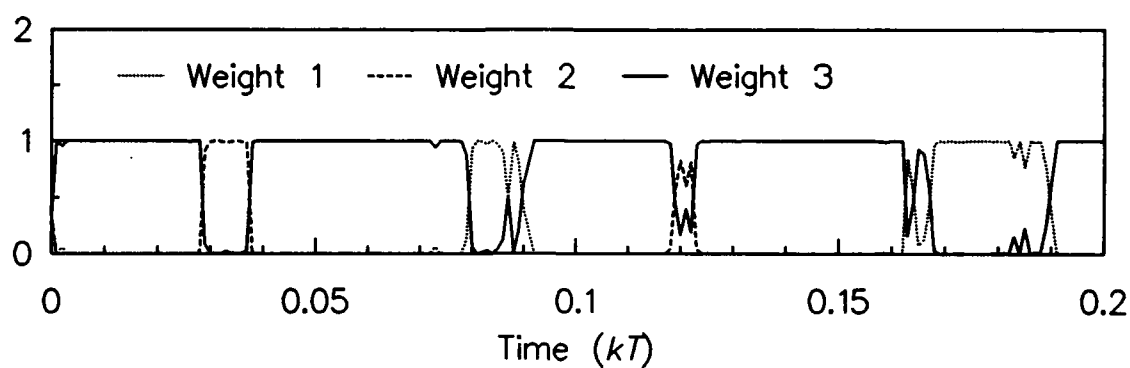


(c)

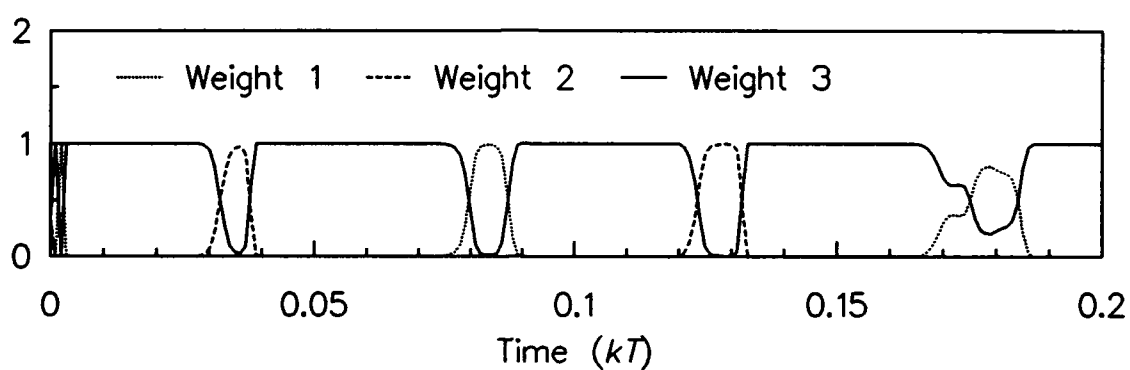


(d)

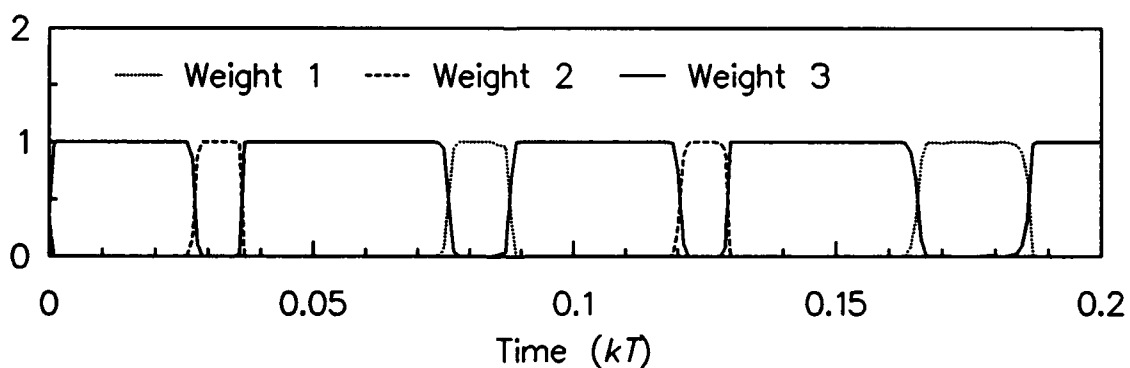
Figure 19. Mode 3 velocity: (a) State and estimate \hat{x}_{k+1}^v , (b) state and estimate $\hat{x}_{k+1}^{\tilde{x}}$, (c) state and estimate $\hat{x}_{k+1}^{\tilde{v}}$, (d) state and estimate $\hat{x}_{k+1}^{\tilde{\tilde{x}}}$.



(a)



(b)



(c)

Figure 20. (a) Weighting terms, estimator *a*, (b) weighting terms, estimators *b* and *c*, (c) weighting terms, estimator *d*.

Table 8.

Normalized mean-square-error percentage for state estimates.

| State | Estimator <i>a</i> \hat{x}_{k+1}^v | Estimator <i>b</i> $\hat{x}_{k+1}^{\tilde{a}}$ | Estimator <i>c</i> $\hat{x}_{k+1}^{\tilde{v}}$ | Estimator <i>d</i> $\hat{x}_{k+1}^{\tilde{c}}$ |
|-----------------|---|---|---|---|
| Mode 1 position | 1.60 | 3.62 | 17.04 | 11.71 |
| Mode 1 velocity | 0.87 | 26.70 | 6.93 | 5.93 |
| Mode 2 position | 1.02 | 33.83 | 15.32 | 12.20 |
| Mode 2 velocity | 1.74 | 11.21 | 6.09 | 4.18 |
| Mode 3 position | 0.94 | 24.73 | 16.70 | 13.06 |
| Mode 3 velocity | 0.39 | 39.03 | 0.34 | 0.38 |

$$\text{mse \%} = \frac{E[(x - \hat{x})^2]}{E[x^2]} \times 100$$

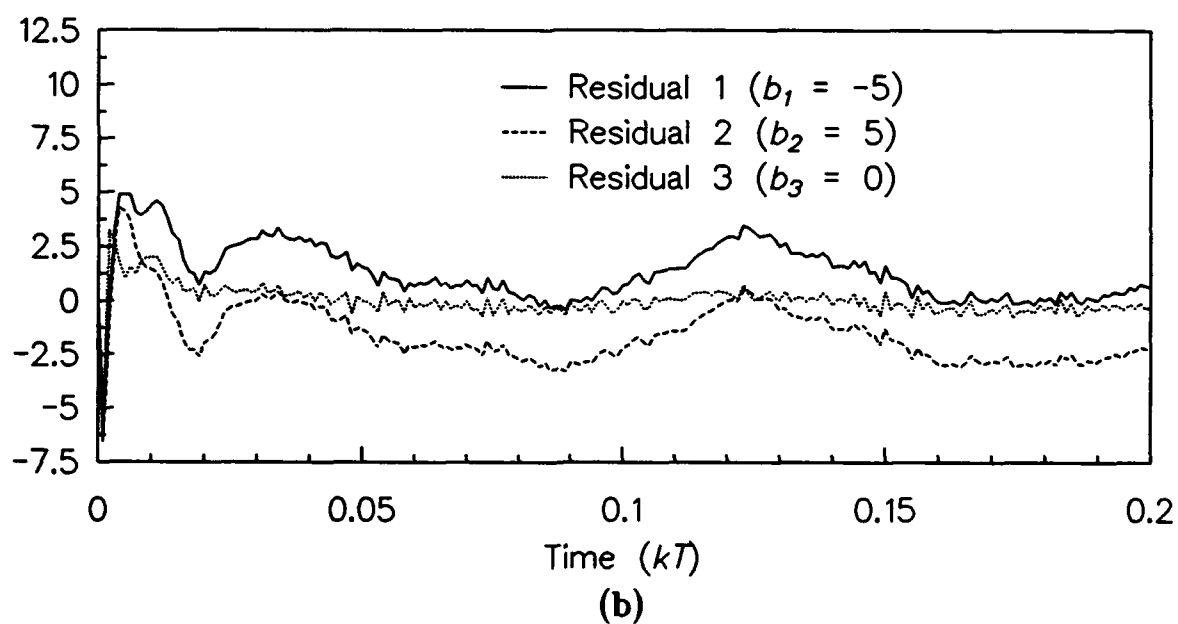
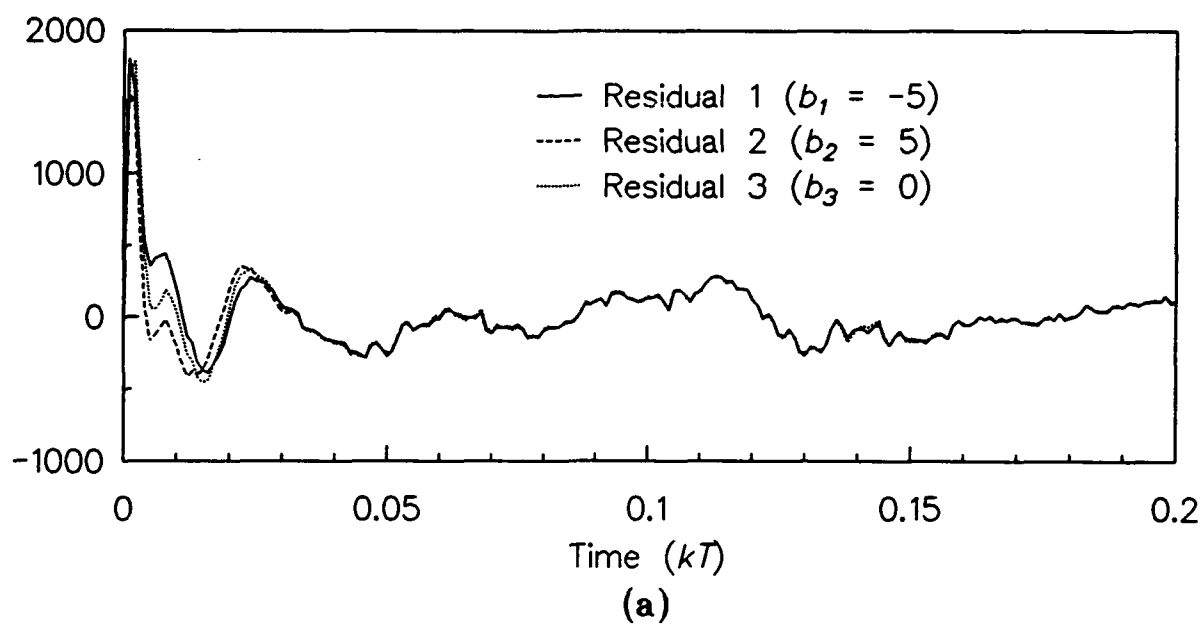


Figure 21. (a) Residuals using the zero dc gain force to acceleration modal system, (b) residuals using the nonzero dc gain force to velocity modal system.

5.0 PARAMETER UPDATING METHODS

As described in chapter 4, the input to the modal system is unknown, but its frequency characteristics are known and the input is always present. It is also not directly measurable by the sensors. The nonGaussian nature of the stochastic FM input signal is modeled using the b_i and S_i parameters of the Gaussian sum signal model. These parameters are closely related to the amplitude of this signal, due to the nature of its probability density function (4.2.10). Two of the b_i bias terms are equal to the positive and negative peak amplitudes of the FM signal, with small corresponding variances. The S_i term associated with the $b_i = 0$ bias term has a value of roughly $(\frac{2}{5}A_u)^2$, where A_u is the peak value of the input.

The stochastic FM signals used in the simulation examples in chapter 4 had constant amplitudes. In practice, the amplitude may shift over a period of time. For good performance under these conditions, the MGS adaptive filter must have some approximate knowledge of the amplitude of the FM signal in order to update the b_i and S_i parameters. Two methods of updating the parameters of the MGS adaptive filter are described in this chapter. One method involves processing the measurement residual of the filter incorporating the $b_i = 0$ (zero-bias) term. The second method involves the use of a Gaussian double-sum to detect a shift in

amplitude. A simulation example is used to examine the performance of these two methods.

5.1 Zero-Bias Measurement Residual Method

One method of updating the b_i and S_i parameters of the MGS adaptive filter involves processing the measurement residual of the filter in the parallel bank of Kalman filters incorporating the $b_i = 0$ (zero-bias) term. Using the modal system model of chapter 4 and the MGS adaptive filter equations of Table 3, the i^{th} measurement residual is

$$r_{k+1}^i = z_{k+1} - Db_i - H(\Phi\hat{x}_k^i + \Gamma b_i) \quad (5.1.1)$$

For notational clarity, set $i = 0$ and $b_0 = 0$ in (5.1.1) to produce the zero-bias measurement residual

$$r_{k+1}^0 = z_{k+1} - H\Phi\hat{x}_k^0 \quad (5.1.2)$$

Substituting measurement equation (4.1.2) at time iteration $k+1$ into (5.1.2) gives

$$r_{k+1}^0 = Hx_{k+1} + Du_{k+1} + v_{k+1} - H\Phi\hat{x}_k^0 \quad (5.1.3)$$

and substituting plant equation (4.1.1) into (5.1.3) gives

$$r_{k+1}^0 = H(\Phi x_k + \Gamma u_k + \Psi w_k) + Du_{k+1} + v_{k+1} - H\Phi\hat{x}_k^0 \quad (5.1.4)$$

Rearranging (5.1.4) produces

$$r_{k+1}^0 = H\Phi(x_k - \hat{x}_k^0) + H\Gamma u_k + Du_{k+1} + H\Psi w_k + v_{k+1} \quad (5.1.5)$$

Now, assuming that the input signal u_k is much larger than both w_k and v_{k+1} such that these two terms can be neglected, assuming that the input signal changes slowly enough such that u_k is approximately equal to u_{k+1} , and setting the error of the zero-bias filter to be $e_k^0 = (x_k - \hat{x}_k^0)$, r_{k+1}^0 of (5.1.5) becomes approximately equal to

$$r_{k+1}^0 \simeq H\Phi e_k^0 + (H\Gamma + D)u_k \quad (5.1.6)$$

Premultiplying both sides of (5.1.6) by the inverse of $(H\Gamma + D)$ if $(H\Gamma + D)$ is square, or by its pseudo-inverse if non-square, and taking the expected value, the zero-bias measurement residual is processed as

$$\tilde{u}_{k+1} = (H\Gamma + D)^{-1}E[r_{k+1}^0] \simeq (H\Gamma + D)^{-1}H\Phi E[e_k^0] + E[u_k] \quad (5.1.7)$$

Assuming that the expected value of the error is $E[e_k^0] = 0$, and given that $E[u_k] = u_k$, the processed zero-bias measurement residual is approximately equal to the input signal

$$\tilde{u}_{k+1} \simeq u_k \quad (5.1.8)$$

Information contained in \tilde{u}_{k+1} is used to periodically update the b_i and S_i parameters of the MGS adaptive filter. The zero-bias residual (5.1.2) is computed

and stored at each iteration. After n iterations, the residual sequence is processed using (5.1.7), producing (5.1.8). The positive and negative peak values of (5.1.8) are found from the sequence, and the power of the sequence is calculated. A combination of these values is used to update the b_i and S_i parameters in order to reduce errors related to the assumptions used in the development of (5.1.8). Due to the sinusoidal nature of the input signal, the b_1 (negative-bias) and b_2 (positive-bias) parameters can be updated using

$$b_1 = \beta P_n - (1 - \beta)\sqrt{2P_{sin}} \quad (5.1.9)$$

$$b_2 = \beta P_p + (1 - \beta)\sqrt{2P_{sin}} \quad (5.1.10)$$

where P_n is the negative peak value of the sequence, P_p is the positive peak value of the sequence, P_{sin} is the power of the sequence, and β is a weighting factor between zero and one. Once b_2 (5.1.10) is updated, the S_3 (variance of the zero-bias filter) parameter is updated using

$$S_3 = (0.4b_2)^2 \quad (5.1.11)$$

After all the parameters are updated, the sequence is cleared and a new sequence is initiated for the next n -sample update.

A simulation example illustrating the use of the zero-bias measurement residual method of updating the b_i and S_i parameters is now given. A modified first-order discrete-time system from [60] is used, given by

$$x_{k+1} = \Phi x_k + \Gamma u_k + \Psi w_k \quad (5.1.12)$$

$$z_k = H x_k + D u_k + v_k \quad (5.1.13)$$

with values

$$\Phi = 0.7 \quad \Gamma = 1.0 \quad \Psi = 1.0 \quad H = 0.5 \quad D = 2.0 \quad (5.1.14)$$

The plant noise w_k is zero-mean white Gaussian with covariance $Q = 0.05$, and is uncorrelated with measurement noise v_k , which is also zero-mean white Gaussian with covariance $R = 0.05$.

The nonGaussian stochastic FM input signal (4.2.9) in this example has an amplitude of $A_u = 5$ for the first 60 iterations and an amplitude of $A_u = 2$ for the remaining 140 iterations. The input is modeled using the Gaussian sum signal model described in section 4.1

$$u_k = b_k + n_k \quad (5.1.15)$$

Three filters of the form found in Table 3 are used in the simulation. The initial b_i and S_i parameters for the three filters are

$$b_1 = -5 \quad b_2 = 5 \quad b_3 = 0 \quad (5.1.16)$$

$$S_1 = 0.04 \quad S_2 = 0.04 \quad S_3 = 4.28 \quad (5.1.17)$$

The MGS adaptive filter is initialized with equally-valued weighting terms

$$w_0^j = \frac{1}{3}, \quad \text{for } j = 1, 2, 3 \quad (5.1.18)$$

A Markov transition probability matrix Θ_{bS}^{ij} , consisting of θ_{bS}^{ij} elements, is configured with a high probability that the bias term does not switch from one

value to another, and a low probability that the bias term does switch, such as

$$\Theta_{bS}^{ij} = \begin{bmatrix} .90 & .05 & .05 \\ .05 & .90 & .05 \\ .05 & .05 & .90 \end{bmatrix} \quad (5.1.19)$$

The initial values of the state, state estimate, and error covariance for the MGS adaptive filter are

$$x_0 = 0 \quad \hat{x}_0 = 16 \quad P_o = 10000 \quad (5.1.20)$$

Figure 22 compares the stochastic FM signal with the processed zero-bias measurement residual (5.1.8). The two are quite similar in appearance. Figure 23a shows the state x_{k+1} and the state estimate \hat{x}_{k+1} without using the parameter updating method. The initial estimates are good since the parameters are properly matched to the input. However, after the input switches amplitude from $A_u = 5$ to $A_u = 2$, the estimates degrade since the parameters are now mismatched. Figure 24a shows the state x_{k+1} and the state estimate \hat{x}_{k+1} using the parameter updating method every 30 iterations. The initial estimates track well. Then the estimator goes through a period of learning a new set of parameter values as the amplitude of the input changes. Once the correct parameters are found, the estimator tracks with minimal error. Figure 24a shows the weighting terms of the MGS adaptive filter with no parameter updating, and Figure 24b shows the weighting terms of the MGS adaptive filter with parameter updating. Note how only the MGS filter with parameter updating develops proper weighting terms for the estimates during the time the signal has an amplitude of $A_u = 2$.

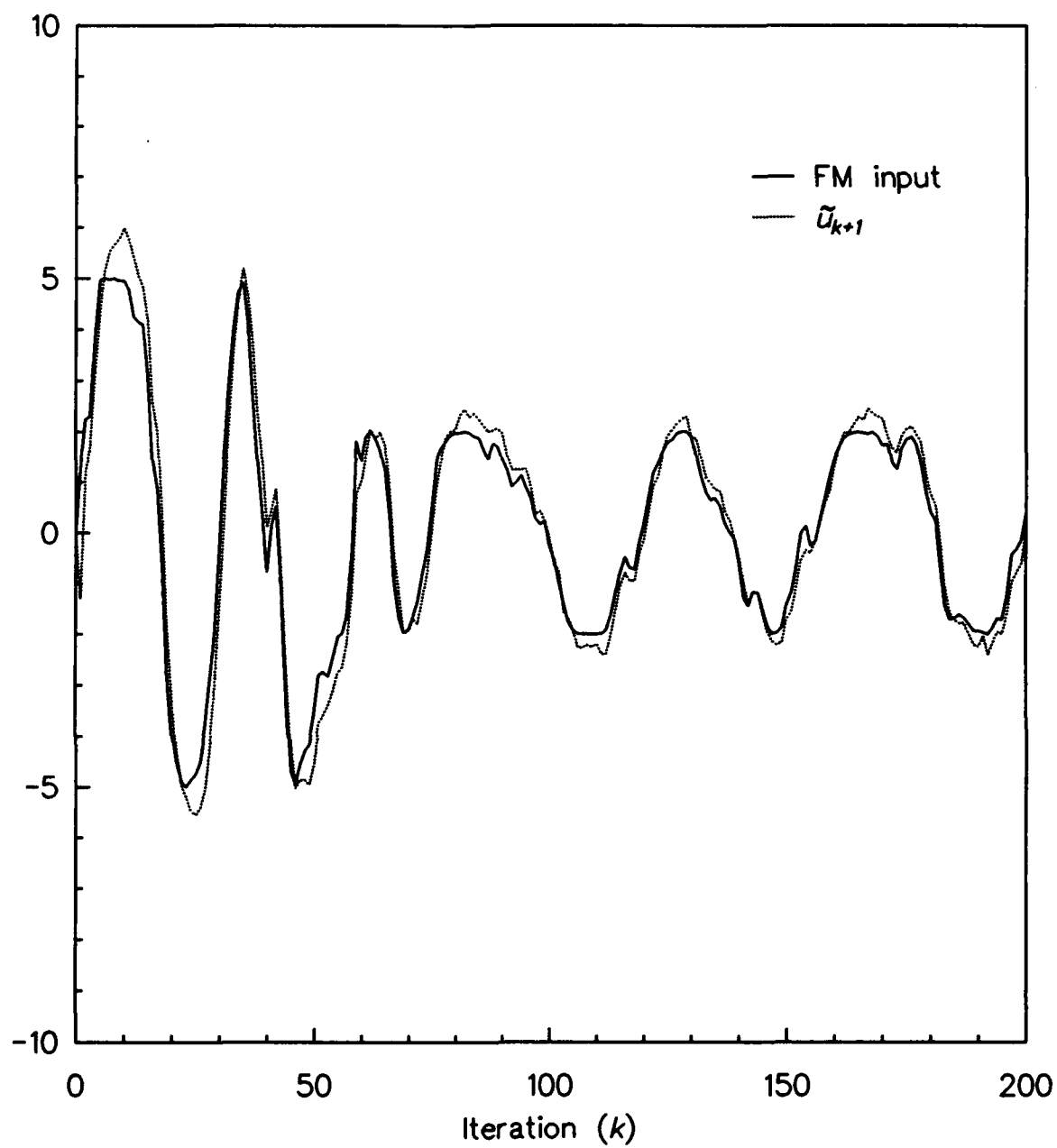


Figure 22. Stochastic FM signal and processed zero-bias measurement residual.

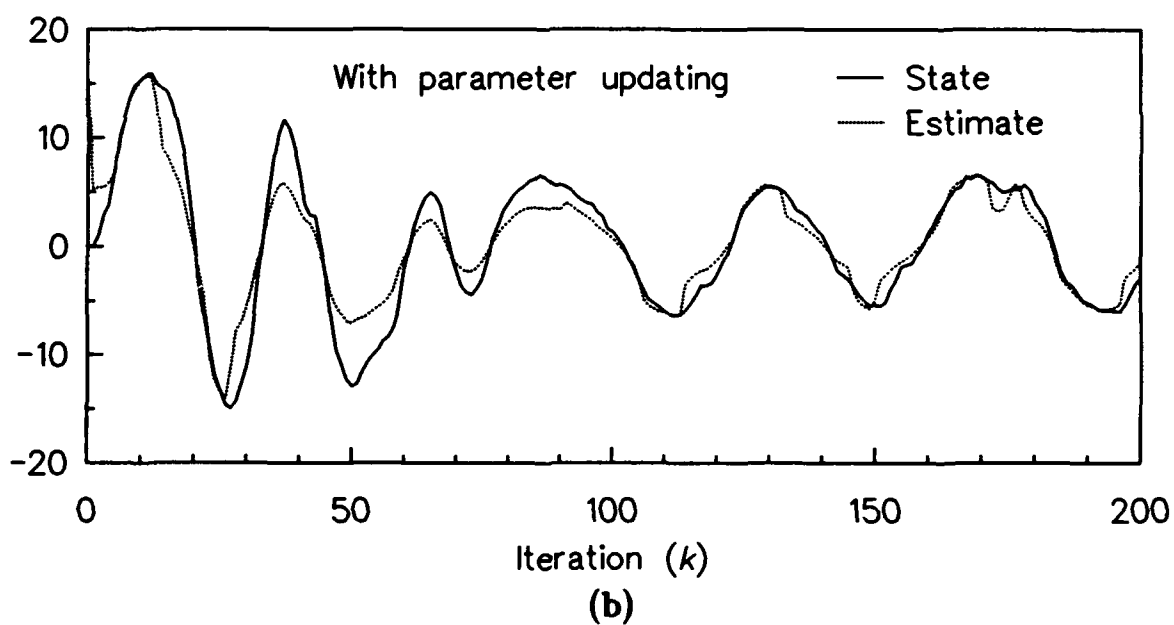
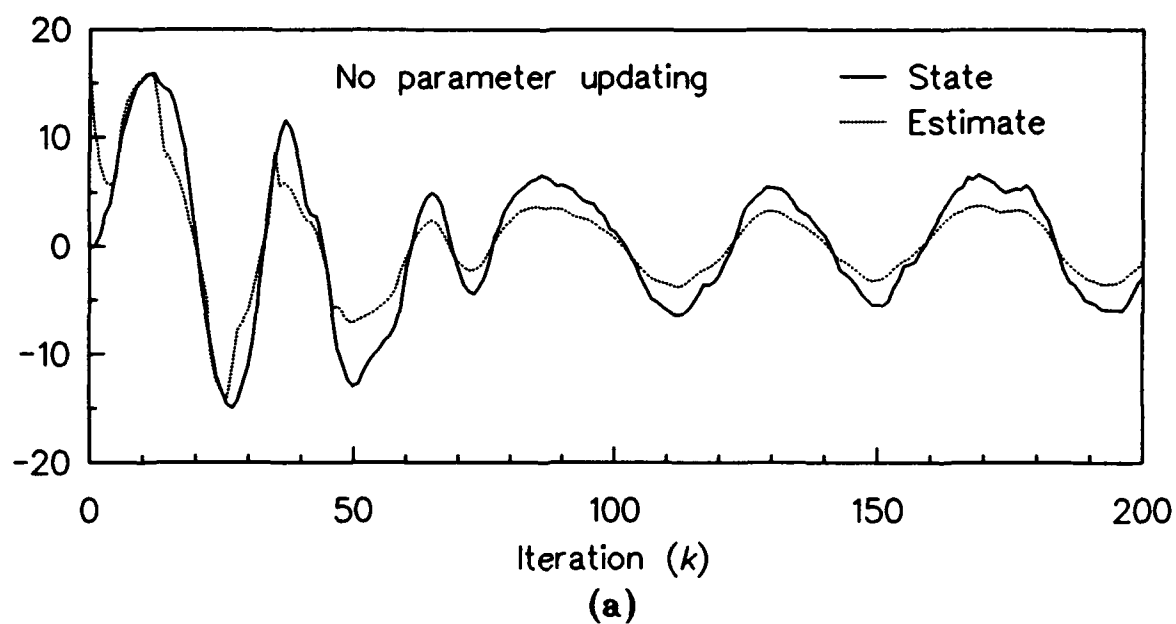


Figure 23. (a) State and state estimate with no parameter updating, (b) state and state estimate using zero-bias measurement residual parameter updating method.

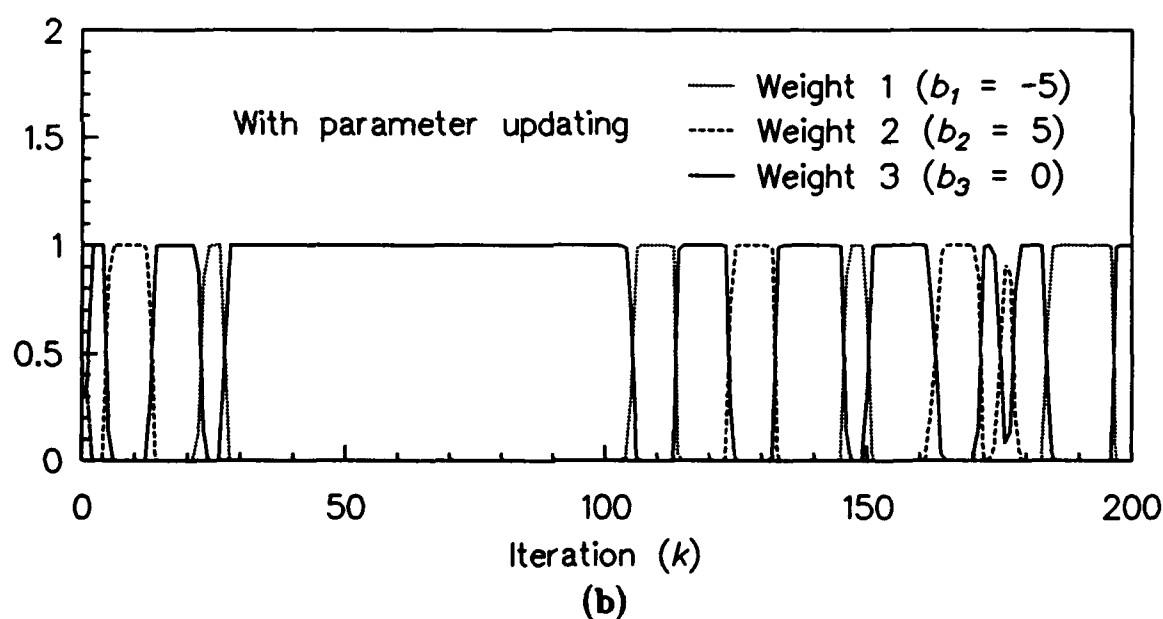
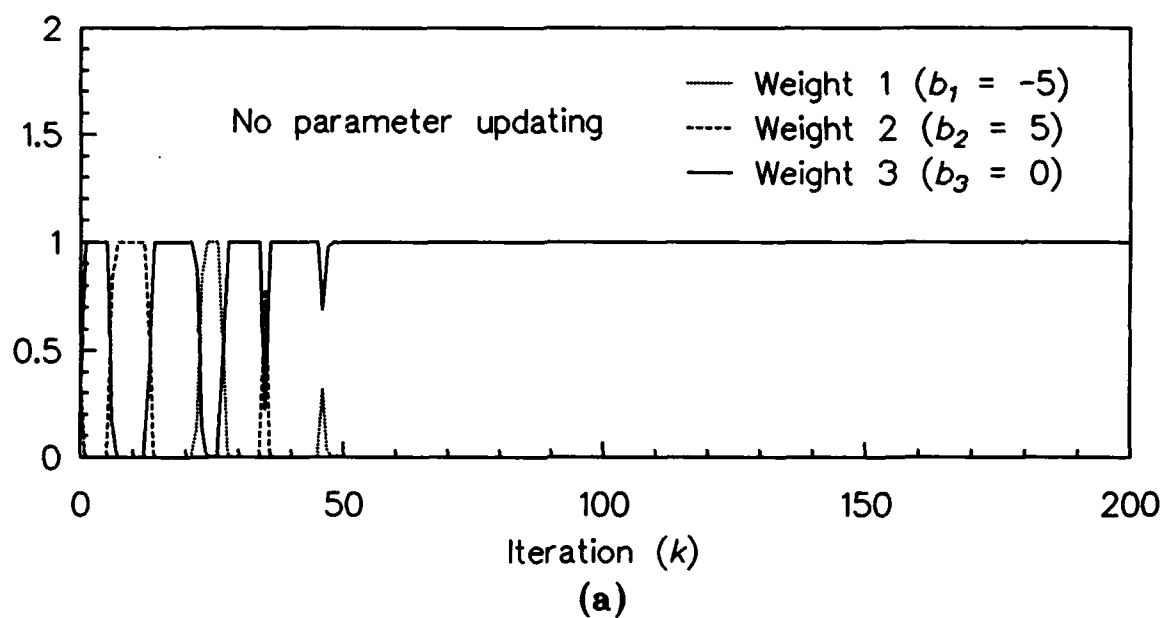


Figure 24. (a) Weighting terms with no parameter updating, (b) weighting terms using zero-bias measurement residual parameter updating method.

5.2 Gaussian Double-Sum Method

A second method of updating the parameters of the MGS adaptive filter involves the use of a Gaussian double-sum to detect a shift in amplitude. The nonGaussian signal model (5.1.12) has a Gaussian sum probability density function of the form

$$p(u) = \sum_{i=1}^M P_i N[b_i, S_i] \quad (5.2.1)$$

Now, a new nonGaussian input signal model is generated by multiplying the original model by a constant gain, producing

$$s = Au = A(b + n) \quad (5.2.2)$$

The gain A is a semi-Markov process with values randomly selected from a fixed set of discrete values, characterized by a delta probability density function

$$p(A) = \sum_{j=1}^N P_j^A \delta(A - A_j) \quad (5.2.3)$$

with

$$\sum_{j=1}^N P_j^A = 1; \quad P_j^A \geq 0 \text{ for } j = 1, 2, \dots, N \quad (5.2.4)$$

This process can be thought of as a randomly-switching gain, each gain value A_j having probability P_j^A . It is assumed that A and u are statistically independent random processes.

The probability density function of the new nonGaussian input signal model (5.2.2) is found using a general result from [61]

$$p(s) = \int_{-\infty}^{\infty} \frac{1}{|A|} p_u(s/A) p_A(A) dA \quad (5.2.5)$$

substituting (5.2.3) into (5.2.5) gives

$$p(s) = \int_{-\infty}^{\infty} \frac{1}{|A|} p_u(s/A) \sum_{j=1}^N P_j^A \delta(A - A_j) dA \quad (5.2.6)$$

Interchanging integration and summation

$$p(s) = \sum_{j=1}^N P_j^A \left[\int_{-\infty}^{\infty} \frac{1}{|A|} p_u(s/A) \delta(A - A_j) dA \right] \quad (5.2.7)$$

Using the sifting property of the delta function [54]

$$f(a) = \int_{-\infty}^{\infty} f(x) \delta(x - a) dx \quad (5.2.8)$$

the integral of (5.2.7) is evaluated as

$$p(s) = \sum_{j=1}^N P_j^A \left[\frac{1}{|A_j|} p_u(s/A_j) \right] \quad (5.2.9)$$

The Gaussian density $N[b_i, S_i]$ of (5.2.1) has the form

$$N[b_i, S_i] = \frac{1}{\sqrt{2\pi S_i}} e^{-\frac{1}{2} \left[\frac{(u - b_i)^2}{S_i} \right]} \quad (5.2.10)$$

Solving (5.2.2) for $u = s/A_j$ and substituting into (5.2.10) produces

$$N[b_i, S_i] = \frac{1}{\sqrt{2\pi S_i}} e^{-\frac{1}{2} \left[\frac{\left(\frac{s}{A_j} - b_i \right)^2}{S_i} \right]} \quad (5.1.11)$$

and rearranging the exponent gives

$$N[b_i, S_i] = \frac{1}{\sqrt{2\pi S_i}} e^{-\frac{1}{2} \left[\frac{(s - A_j b_i)^2}{A_j^2 S_i} \right]} \quad (5.2.12)$$

Combining (5.2.12) and (5.2.1) and substituting into (5.2.9) produces

$$p(s) = \sum_{i=1}^M P_i \sum_{j=1}^N P_j^A \frac{1}{\sqrt{2\pi A_j^2 S_i}} e^{-\frac{1}{2} \left[\frac{(s - A_j b_i)^2}{A_j^2 S_i} \right]} \quad (5.2.13)$$

or in abbreviated notation

$$p(s) = \sum_{i=1}^M \sum_{j=1}^N P_i P_j^A N[A_j b_i, A_j^2 S_i] \quad (5.2.14)$$

Thus, the nonGaussian density function of s can be modeled as the Gaussian double-sum (5.2.14). The A_j parameter is used to account for any shift in the amplitude of the actual nonGaussian input signal. Several values of A_j are selected to cover a range of possible input amplitudes. If the actual amplitude changes, the MGS algorithm will react by selecting the filters using the most properly matched A_j value. However, a total of MN filters is now required to implement the MGS

algorithm, rather than only M filters previously required. Following a similar development given in chapter 3, the modified Gaussian double-sum (MGDS) adaptive filter equations for a modal system are summarized in Table 9. A block diagram of the MGDS adaptive filter is shown in Figure 25.

The simulation example of section 5.1 will be used again, this time illustrating the use of the Gaussian double-sum method of updating the MGS adaptive filter parameters. Three b_i and S_i parameter values are used again

$$b_1 = -1 \quad b_2 = 1 \quad b_3 = 0 \quad (5.2.15)$$

$$S_1 = 0.04 \quad S_2 = 0.04 \quad S_3 = 0.16 \quad (5.2.16)$$

In effect, these parameters remain fixed. Choosing the proper A_j parameter causes the MGDS adaptive filter values to change. The S_3 parameter is set to $(\frac{2}{5})^2 = 0.16$ to allow for the correct scaling when multiplied by A_j^2 . The S_1 and S_2 values are not scaled by A_j^2 since they model the steep peaks at either end of the stochastic FM probability density function (4.2.10). These two parameters properly model these peaks for all amplitude values.

Two A_j values are used in the MGDS adaptive filter to cover the range of the input signal amplitudes used in the simulation. They are

$$A_1 = 2 \quad A_2 = 5 \quad (5.2.17)$$

The MGDS adaptive filter is initialized with equally-valued weighting terms

$$w_0^{ij} = \frac{1}{6}, \quad \text{for } i = 1, 2, 3 \text{ and } j = 1, 2 \quad (5.2.18)$$

Table 9.

Modified Gaussian double-sum adaptive filter equations for a modal system.

System:

$$\begin{aligned}x_{k+1} &= \Phi x_k + \Gamma s_k + \Psi w_k \\z_k &= H x_k + D s_k + v_k \\s_k &= A_k(b_k + n_k)\end{aligned}$$

Overall estimate:

$$\hat{x}_{k+1} = \sum_{i=1}^M \sum_{j=1}^N \hat{x}_{k+1}^{ij} p(b_i, S_i, A_j | Z_{k+1})$$

Kalman filter equation:

$$\hat{x}_{k+1}^{ij} = \Phi \hat{x}_k^{ij} + \Gamma A_j b_i + K_{k+1}^{ij} [z_{k+1} - D A_j b_i - H(\Phi \hat{x}_k^{ij} + \Gamma A_j b_i)]$$

Kalman gain equations:

$$M_{k+1}^{ij} = \Phi P_k^{ij} \Phi^T + \Gamma A_j^2 S_i \Gamma^T + \Psi Q \Psi^T$$

$$K_{k+1}^{ij} = M_{k+1}^{ij} H^T (H M_{k+1}^{ij} H^T + D A_j^2 S_i D^T + R)^{-1}$$

$$P_{k+1}^{ij} = (I - K_{k+1}^{ij} H) M_{k+1}^{ij}$$

Weighting term:

$$w_{k+1}^{ij} = p(b_i, S_i, A_j | Z_{k+1}) = C_{k+1}^o p(z_{k+1} | b_i, S_i, A_j, Z_k) \sum_{\alpha=1}^M \sum_{\beta=1}^N \theta_{bS}^{i\alpha} \theta_A^{j\beta} w_k^{\alpha\beta}$$

with

$$p(z_{k+1} | b_i, S_i, A_j, Z_k) = N[H(\Phi \hat{x}_k^{ij} + \Gamma A_j b_i) + D A_j b_i, H M_{k+1}^{ij} H^T + D A_j^2 S_i D^T + R]$$

$$\theta_{bS}^{i\alpha} = p(b_{k+1} = b_i, S_{k+1} = S_i | b_k = b_\alpha, S_k = S_\alpha)$$

$$\theta_A^{j\beta} = p(A_{k+1} = A_j | A_k = A_\beta)$$

$$w_k^{\alpha\beta} = p(b_k = b_\alpha, S_k = S_\alpha, A_k = A_\beta | Z_k)$$

and scale factor C_{k+1}^o such that $\sum_{i=1}^M \sum_{j=1}^N w_{k+1}^{ij} = 1$

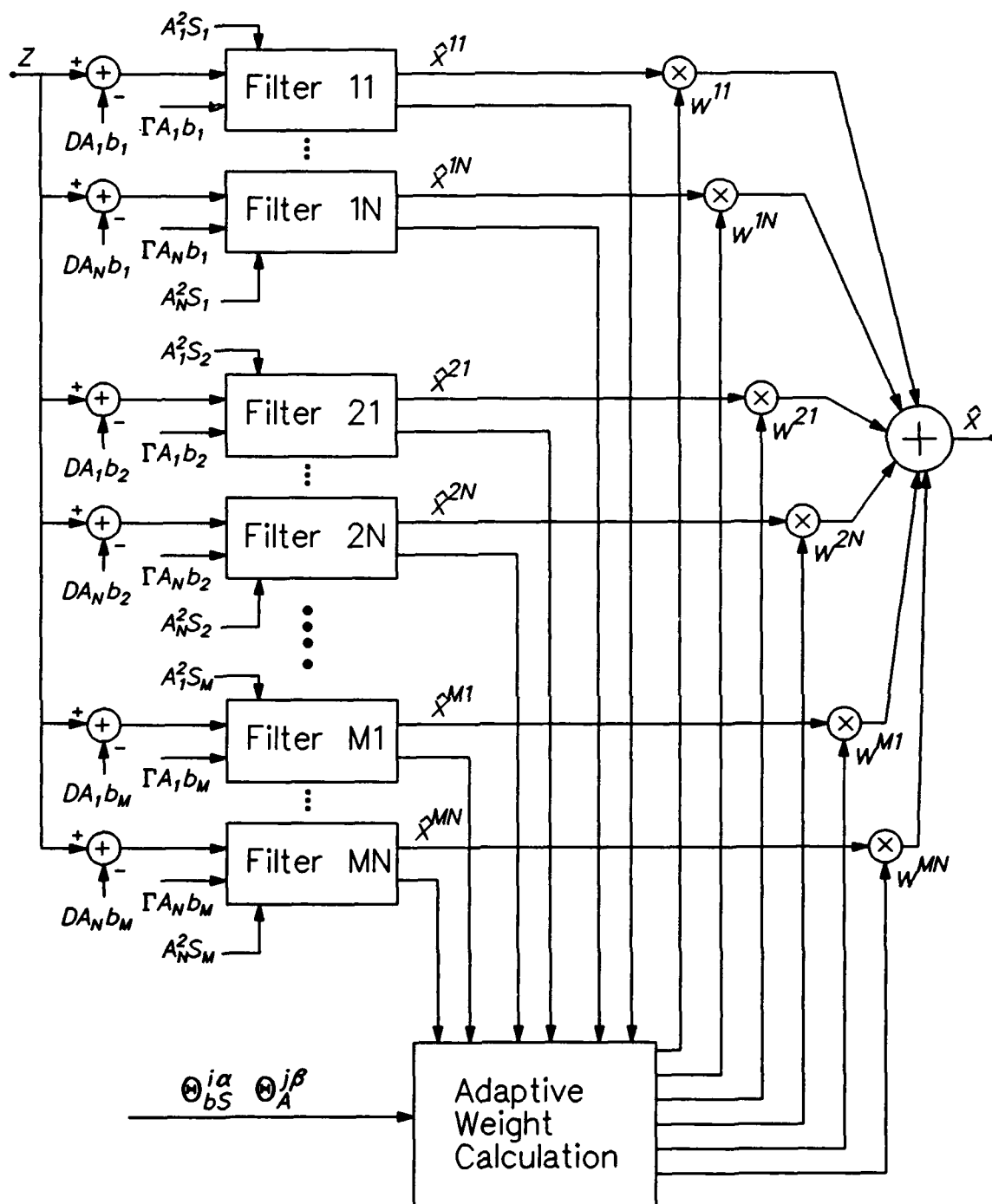


Figure 25. Modified Gaussian double-sum adaptive filter structure.

Two Markov transition probability matrices, $\Theta_{bS}^{i\alpha}$ and $\Theta_A^{j\beta}$, are used in the MGS adaptive filter, given by

$$\Theta_{bS}^{i\alpha} = \begin{bmatrix} .90 & .05 & .05 \\ .05 & .90 & .05 \\ .05 & .05 & .90 \end{bmatrix} \quad \Theta_A^{j\beta} = \begin{bmatrix} .95 & .05 \\ .05 & .95 \end{bmatrix} \quad (5.2.19)$$

Figure 26a shows the state x_{k+1} and the same state estimate \hat{x}_{k+1} generated previously without parameter updating. Figure 26b shows the state x_{k+1} and the state estimate \hat{x}_{k+1} using the Gaussian double-sum parameter updating method. The MGDS estimates track well throughout the simulation, without any noticeable learning time.

Figure 27a shows weighting terms 1-3 of the MGDS adaptive filter with $A_j = 2$, and Figure 27b shows weighting terms 4-6 with $A_j = 5$. The input signal starts out with an amplitude of $A_u = 5$, so weighting terms 4-6 initially are in effect. After the amplitude switches to $A_u = 2$, weighting terms 1-3 take over, with one exception. Weight 3 (for $b_i = 0$) should be larger than weight 6 (also for $b_i = 0$) since $A_j = 2$ currently matches the actual input amplitude. However, the measurement residuals for corresponding filters 3 and 6 are equal, because their b_i terms are each zero. This causes the difference in their S_i variance terms to be the deciding factor in calculating the weights. The Gaussian density function in Table 9 used to calculate the weights has a larger variance for weight 6 ($A_j = 5$) than for weight 3 ($A_j = 2$). Therefore, the probability calculated for weight 6 will be larger than the probability calculated for weight 3, so that weight 6 governs the overall state estimate more so than weight 3.

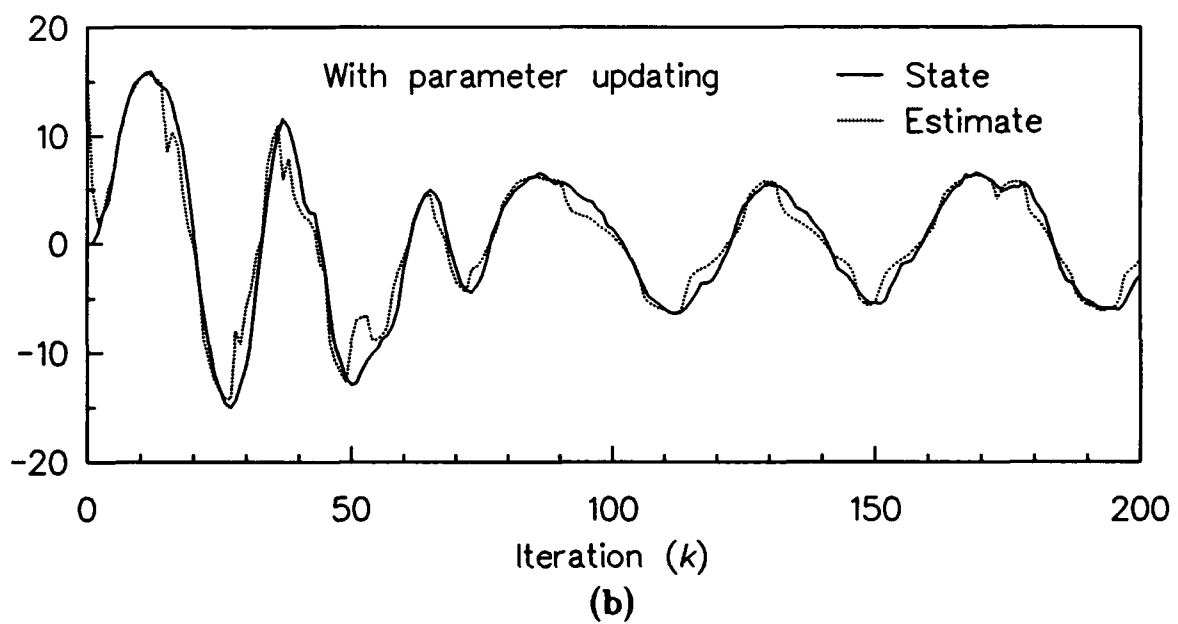
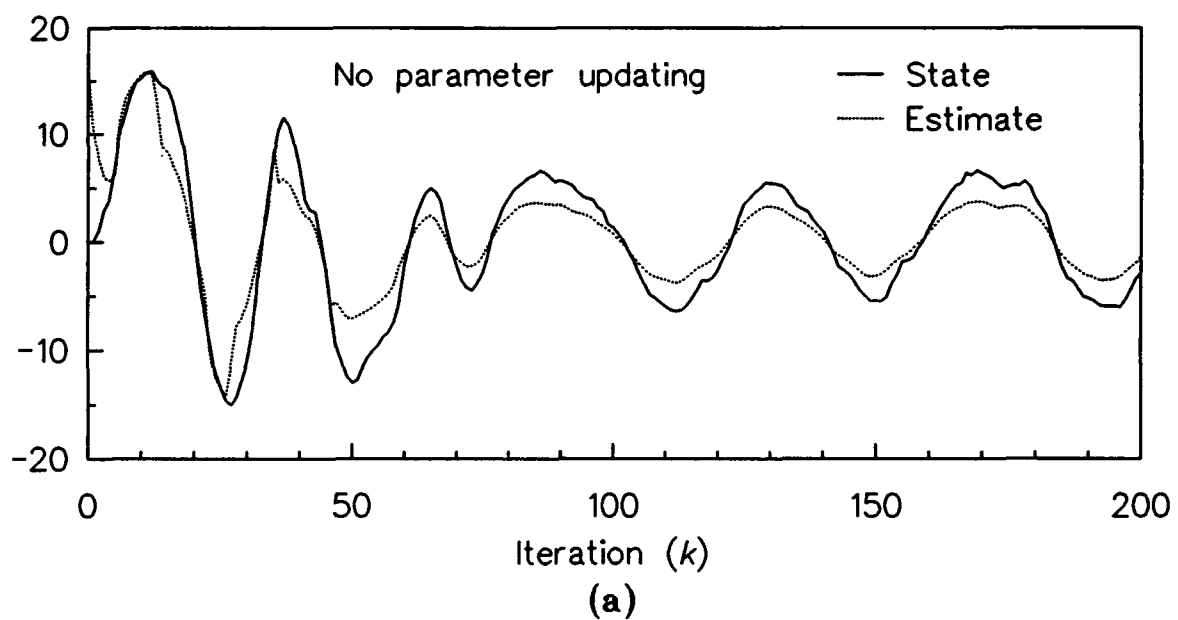


Figure 26. (a) State and state estimate with no parameter updating, (b) state and state estimate using Gaussian double-sum parameter updating method.

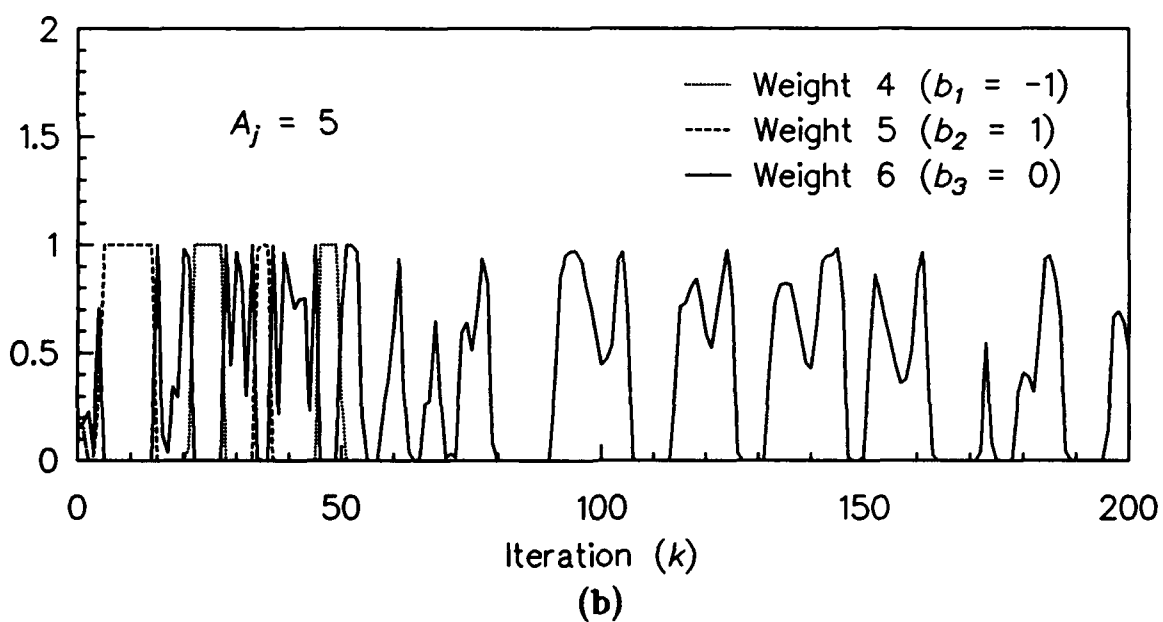
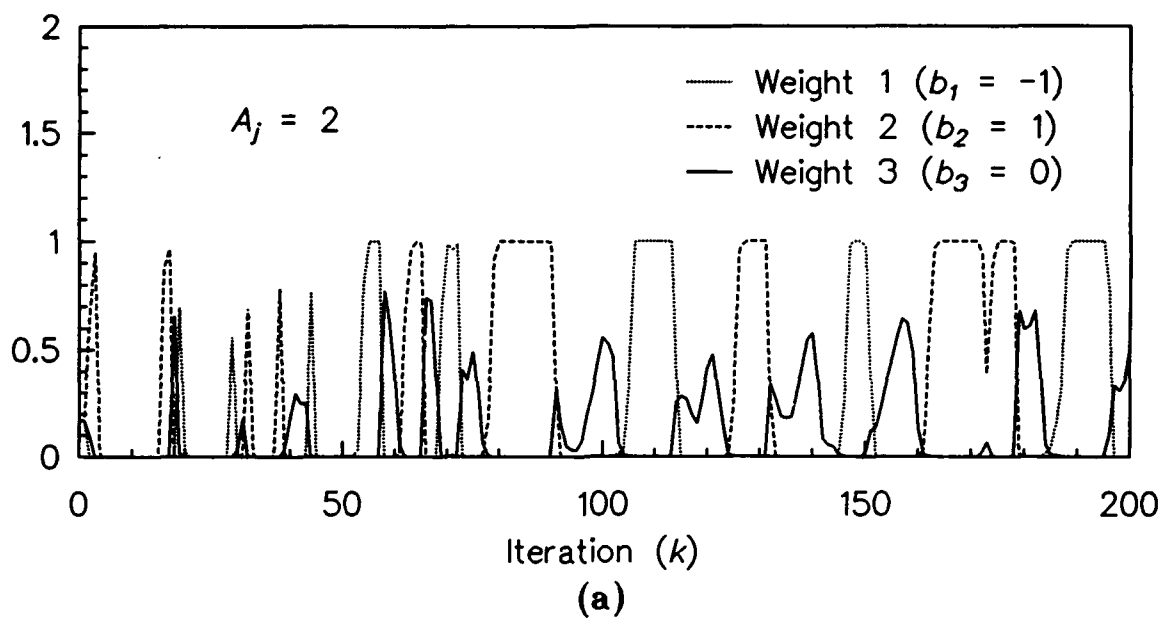


Figure 27. (a) Weighting terms 1-3 with $A_j = 2$,
 (b) weighting terms 4-6 with $A_j = 5$.

6.0 CONCLUSIONS

This investigation is concerned with effective state estimation of a system driven by an unknown nonGaussian input with additive white Gaussian noise, and observed by measurements containing feedthrough of the same nonGaussian input and corrupted by additional white Gaussian noise. A Gaussian sum (GS) approach has previously been developed [6-8] which can cope with the nonGaussian nature of the input signal. Due to a serious growing memory problem in this approach, a modified Gaussian sum (MGS) estimation technique is developed that avoids the growing memory problem while providing effective state estimation. Several differences between the MGS and GS algorithms are examined, showing the MGS algorithm to be a better performer.

An MGS adaptive filter is derived for a general system and a modal system, with simulation examples performed using a nonGaussian input signal. The modal system simulation results are compared to those produced from an augmented Kalman filter based on an augmented modal system model assuming a narrowband Gaussian input signal. The results show that the MGS adaptive filter provides better estimates than the augmented Kalman filter.

A necessary condition for effective MGS estimation is derived, namely that the

system must have a nonzero dc gain. Three alternate estimation procedures are developed to compensate for situations when this condition is not satisfied. The alternate MGS adaptive filters are simulated and their performance results are analyzed and compared.

Two methods of monitoring and updating key parameters of the MGS filter are developed. One is the zero-bias measurement residual method and the second is the Gaussian double-sum method. Simulation results are analyzed to investigate the performance of these methods, with the Gaussian double-sum method proving to be a better performer at the expense of increased computational burden.

Several contributions to the field of applied estimation theory are made from this investigation. These include:

1. The development of a modified Gaussian sum algorithm with nongrowing memory based on a nonGaussian signal model with a Gaussian sum probability density function. Parameters from this model are used directly in the modified Gaussian sum adaptive filter structure.
2. A comparison between the GS filter of [6] and the MGS adaptive filter. The two are similar, but the comparison shows the MGS adaptive filter to be a good improvement to the GS filter.
3. An examination of a necessary condition for effective MGS estimation. This condition provides a simple test to determine if the MGS adaptive filter will work properly for a given system.
4. An alternate configuration of the MGS adaptive filter when the necessary

condition of 3 above is not met. This configuration is applied in several ways, and each is evaluated on a performance basis.

5. Two methods of monitoring and updating key parameters of the MGS adaptive filter. These allow the estimator to react to changes in the input signal level which cause the signal to be nonstationary over long periods of time.

Some suggested directions for future development include a closer examination of the assumptions used in the MGS algorithm to alleviate the increasing computational and storage requirements of the growing memory GS algorithm, applying the MGS algorithm to a larger variety of nonGaussian input signals, and examining how to implement the algorithm in the case of nonlinear systems with nonGaussian inputs.

APPENDIX A

The transfer matrix from actuator force inputs to sensor acceleration outputs for the modal model described in [1] is

$$G(s) = s \sum_{i=1}^m \left[\frac{c_i \phi_i \phi_i^T}{s - \lambda_i} + \frac{c_i^* \phi_i^* \phi_i^{*T}}{s - \lambda_i^*} \right] \quad (\text{A.1})$$

where

m is the total number of modeled (complex) modes

λ_i is the i^{th} eigenvalue of the transfer matrix

ϕ_i is the i^{th} mode shape vector (eigenvector)

c_i is the i^{th} modal participation factor

An equivalent continuous-time state representation of (A.1) has the form

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (\text{A.2})$$

$$y(t) = Cx(t) + Du(t) \quad (\text{A.3})$$

where

$$x(t) \equiv \begin{bmatrix} q_1(t) & \dot{q}_1(t) & \cdots & q_m(t) & \dot{q}_m(t) \end{bmatrix}^T \quad (\text{A.4})$$

are the modal positions and velocities, $u(t)$ is the m -length vector of actuator input forces, and $y(t)$ is the m -length vector of accelerations at sensor grid points.

Clearly, the dc gain of (A.1) is zero since for $s = 0$, $G(s) = 0$. Differentiating (A.1) once with respect to s removes the s in front of the summation, producing a new transfer matrix from actuator force inputs to sensor velocity outputs. This new transfer matrix has a nonzero dc gain. The discrete-time equations (4.1.1) and (4.1.2) are generated using a zero-order-hold model. Therefore, the discrete-time dc gain (4.2.4) from u_k to z_k is equal to the continuous-time dc gain.

The A , B , C , and D matrices of the state space representation of the force to acceleration transfer matrix are computed as follows. Define

$$\begin{aligned} \Lambda &\equiv \text{diag}\{\lambda_1, \lambda_1^*, \dots, \lambda_m, \lambda_m^*\} \\ \mathcal{C} &\equiv \text{diag}\{c_1, c_1^*, \dots, c_m, c_m^*\} \\ \Phi &\equiv \begin{bmatrix} \phi_1 & \phi_1^* & \cdots & \phi_m & \phi_m^* \end{bmatrix} \end{aligned}$$

Then

$$A = \Lambda, \quad B = \Lambda \mathcal{C}^{\frac{1}{2}} \Phi^T, \quad C = \Phi \mathcal{C}^{\frac{1}{2}}, \quad D = \Phi \mathcal{C} \Phi^T \quad (\text{A.5})$$

The A , B , C , and D matrices of the state space representation of the force to velocity transfer matrix are

$$A = \Lambda, \quad B = \Lambda \mathcal{C}^{\frac{1}{2}} \Phi^T, \quad C = \Phi \mathcal{C}^{\frac{1}{2}} \Lambda^{-1}, \quad D = 0 \quad (\text{A.6})$$

The A and B matrices of (A.6) are the same as those from (A.5). This allows the states (A.4) to remain unchanged when either acceleration or velocity outputs are used.

Numerical values for the parameters in (A.5) and (A.6) are generated using the algorithm below. In these equations the subscript i represents the corresponding mode number, and values are generated for $i = 1, 2, \dots, m$. The eigenvalues are given by

$$\lambda_i = -\alpha_i \pm j\omega_i \quad (\text{A.7})$$

where the α_i are randomly selected with uniform distributions over three intervals: [1,2], [2,5], and [5,15], generally with the highest frequency of occurrence in [1,2], followed by [5,15], and then by [2,5]. The ω_i are given by

$$\omega_i = 2\pi f_i \quad (\text{A.8})$$

$$f_i = 100 e^{-0.1(\alpha_i - 1)} \quad (\text{A.9})$$

The eigenfunctions are given by

$$\phi_i(x) = \frac{e^{jk_i x}}{\sqrt{m}} \quad (\text{A.10})$$

where x is the $m \times 1$ vector of grid point locations distributed over a length of 70 meters, and

$$k_i = \frac{\omega_i}{v_i} \quad (\text{A.11})$$

$$v_i = 3000 e^{-0.025 f_i} \quad (\text{A.12})$$

The magnitudes of the modal participation factors are given by

$$|c_i| = 100 e^{.01(f_i - 20)} \quad (\text{A.13})$$

and the phase angles are

$$\angle c_i = \text{random values, uniformly distributed on the interval } [0, 2\pi] \quad (\text{A.14})$$

Using the algorithm above, the parameters of (A.5) and (A.6) are completely specified by α_i , x , and $\angle c_i$ for $i = 1, 2, \dots, m$ and some positive integer m .

LIST OF REFERENCES

- [1] R. L. Moose, H. F. VanLandingham, M. J. Caputi, S. H. Jones, and E. N. Khoury, "Modal control of damped structures," Annu. Rep., ONR Contr. N00014-88-C-0586, Mar. 1990.
- [2] R. E. Kalman, "A new approach to linear filtering and prediction problems," *ASME Trans., J. Basic Eng.*, series 82D, pp. 35-45, Mar. 1960.
- [3] R. E. Kalman and R. S. Bucy, "New results in linear filtering and prediction theory," *ASME Trans., J. Basic Eng.*, series 83D, pp. 95-108, Mar. 1961.
- [4] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1979.
- [5] R. F. Stengel, *Stochastic Optimal Control*. New York: John Wiley and Sons, Inc., 1986.
- [6] D. L. Alspach, "A Bayesian approximation technique for estimation and control of time-discrete stochastic systems," Ph.D. dissertation, Univ. Calif., San Diego, 1970.
- [7] H. W. Sorenson and D. L. Alspach, "Recursive Bayesian estimation using Gaussian sums," *Automatica*, vol. 7, pp. 465-479, 1971.
- [8] D. L. Alspach and H. W. Sorenson, "Nonlinear Bayesian estimation using Gaussian sum approximations," *IEEE Trans. Automat. Contr.*, vol. AC-17, no. 4, pp. 439-448, Aug. 1972.
- [9] D. L. Alspach, "A parallel processing solution to the adaptive Kalman filtering problem with vector measurements," *Comput. & Elect. Engng.*, vol. 1, pp. 83-94, 1973.

- [10] D. L. Alspach, "A parallel filtering algorithm for linear systems with unknown time varying noise statistics," *IEEE Trans. Automat. Contr.*, vol. AC-19, no. 5, pp. 552-556, Oct. 1974.
- [11] D. L. Alspach, L. L. Scharf, and A. Abiri, "A Bayesian solution to the problem of state estimation in an unknown noise environment," *Int. J. Control*, vol. 19, no. 2, pp. 265-287, 1974.
- [12] D. L. Alspach and J. LaGrotta, "A parallel processing Gaussian sum approach to the Dayton problem," *Proc. 4th Symp. Nonlinear Estimation Theory and Its Applications*, San Diego, CA., pp. 1-16, 1973.
- [13] D. L. Alspach, "A Gaussian sum approach to the multi-target identification-tracking problem," *Automatica*, vol. 11, pp. 285-296, 1975.
- [14] D. L. Alspach and H. W. Sorenson, "Stochastic optimal control for linear but non-Gaussian systems," *Int. J. Control*, vol. 13, no. 6, pp. 1169-1181, 1971.
- [15] D. L. Alspach, "Dual control based on approximate *a posteriori* density functions," *IEEE Trans. Automat. Contr.*, vol. AC-17, no. 5, pp. 689-693, Oct. 1972.
- [16] D. T. Magill, "Optimal adaptive estimation of sampled stochastic processes," *IEEE Trans. Auto. Cont.*, vol. AC-10, no. 4, pp. 434-439, Oct. 1965.
- [17] G. A. Ackerson and K. S. Fu, "On state estimation in switching environments," *IEEE Trans. Automat. Contr.*, vol. AC-15, no. 15, pp. 10-17, Feb. 1970.
- [18] A. G. Jaffer and S. C. Gupta, "Optimal sequential estimation of discrete processes with Markov interrupted observations," *IEEE Trans. Automat. Contr.*, vol. AC-16, no. 5, pp. 471-475, Oct. 1971.
- [19] S. Fujita and T. Fukao, "Optimal stochastic control for discrete-time linear system with interrupted observations," *Automatica*, vol. 8, pp. 425-432, 1972.
- [20] J. M. H. Bruckner, R. W. Scott, and F. G. Rea, "Analysis of multimodal systems," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-9, no. 6, pp. 883-888, Nov. 1973.
- [21] R. A. Howard, "System analysis of semi-Markov processes," *IEEE Trans. Mil. Electron.*, vol. MIL-8, pp. 114-124, Apr. 1964.

- [22] R. A. Howard, *Dynamic Probabilistic Systems, Volume II: Semi-Markov and Decision Processes*. New York: John Wiley and Sons, Inc., 1971.
- [23] R. L. Moose, "An adaptive estimator with learning for a plant containing semi-Markov switching parameters," Ph.D. dissertation, Duke Univ., 1971.
- [24] R. L. Moose and P. P. Wang, "An adaptive estimator with learning for a plant containing semi-Markov switching parameters," *IEEE Trans. on Sys., Man, and Cyber.*, vol. SMC-3, no. 3, pp. 277-281, May 1973.
- [25] R. L. Moose, "An adaptive state estimation solution to the maneuvering target problem," *IEEE Trans. Automat. Contr.*, vol. AC-20, no. 3, pp. 359-362, June 1975.
- [26] N. H. Gholson and R. L. Moose, "Maneuvering target tracking using adaptive state estimation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-13, no. 3, pp. 310-317, May 1977.
- [27] R. L. Moose, H. F. VanLandingham, and D. H. McCabe, "Modeling and estimation for tracking maneuvering targets," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-15, no. 3, pp. 448-456, May 1979.
- [28] D. H. McCabe and R. L. Moose, "Passive source tracking using sonar time delay data," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, no. 3, pp. 614-617, June 1981.
- [29] R. L. Moose and T. E. Dailey, "Adaptive underwater target tracking using passive multipath time-delay measurements," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, no. 4, pp. 777-787, Aug. 1985.
- [30] R. L. Moose and P. M. Godiwala, "Passive depth tracking of underwater maneuvering targets," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, no. 4, pp. 1041-1044, Aug. 1985.
- [31] R. L. Moose, M. K. Sistanizadeh, and G. Skagfjord, "Adaptive estimation for a system with unknown measurement bias," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-22, no. 6, pp. 732-738, Nov. 1986.
- [32] R. L. Moose, "Passive range estimation of an underwater maneuvering target," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-35, no. 3, pp. 274-285, Mar. 1987.
- [33] J. K. Tugnait and A. H. Haddad, "A detection-estimation scheme for state estimation in switching environments," *Automatica*, vol. 15, pp. 477-481, 1979.

- [34] J. K. Tugnait and A. H. Haddad, "Adaptive estimation in linear systems with unknown Markovian noise statistics," *IEEE Trans. Inform. Theory*, vol. IT-26, no. 1, pp. 66-78, Jan. 1980.
- [35] H. Akash and H. Kumamoto, "Random sampling approach to state estimation in switching environments," *Automatica*, vol. 13, pp. 429-434, 1977.
- [36] P. K. S. Tam and J. B. Moore, "A Gaussian sum approach to phase and frequency estimation," *IEEE Trans. Comm.*, vol. COM-25, no. 9, pp. 935-942, Sept. 1977.
- [37] S. P. Dmitriev and L. I. Shimelevich, "Generalized Kalman filter with repeated linearization and its application in geophysical field navigation," *Automation & Remote Control*, vol. 39, no. 4, pt. 1, pp. 505-509, Apr. 1978.
- [38] H. R. Sirisena and E. P. M. Brown, "Representation of non-Gaussian probability distributions in stochastic load-flow studies by the method of Gaussian sum approximations," *IEE Proc.*, vol. 130, pt. C, no. 4, pp. 165-171, July 1983.
- [39] T. Namera and A. R. Stubberud, "Gaussian sum approximation for non-linear fixed-point prediction," *Int. J. Control*, vol. 38, no. 5, pp. 1047-1053, 1983.
- [40] M. Gauvrit, "Bayesian adaptive filter for tracking with measurements of uncertain origin," *Automatica*, vol. 20, no. 2, pp. 217-224, 1984.
- [41] M. Tanaka and T. Katayama, "Robust Kalman filter for linear discrete-time system with Gaussian sum noises," *Int. J. Systems Sci.*, vol. 18, no. 9, pp. 1721-1731, 1987.
- [42] G. Kitagawa, "Non-Gaussian seasonal adjustment," *Computers Math. Applic.*, vol. 18, no. 6/7, pp. 503-514, 1989.
- [43] C. J. Masreliez, "Approximate non-Gaussian filtering with linear state and observation relations," *IEEE Trans. Automat. Contr.*, vol. AC-20, no. 1, pp. 107-110, Feb. 1975.
- [44] C. J. Masreliez and R. D. Martin, "Robust Bayesian estimation for the linear model and robustifying the Kalman filter," *IEEE Trans. Automat. Contr.*, vol. AC-22, no. 3, June 1977.
- [45] C. Tsai and L. Kurz, "An adaptive robustizing approach to Kalman filtering," *Automatica*, vol. 19, no. 3, pp. 279-288, 1983.

- [46] P. S. Maybeck and R. I. Suizu, "Adaptive tracker field-of-view variation via multiple model filtering," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-21, no. 4, pp. 529-538, July 1985.
- [47] D. M. Tobin and P. S. Maybeck, "Enhancements to a multiple model adaptive estimator/image-tracker," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-24, no. 4, pp. 417-426, July 1988.
- [48] H. A. P. Blom and Y. Bar-Shalom, "The interacting multiple model algorithm for systems with Markovian switching coefficients," *IEEE Trans. Automat. Contr.*, vol. 33, no. 8, Aug. 1988.
- [49] Y. Bar-Shalom, K. C. Chang, and H. A. P. Blom, "Tracking a maneuvering target using input estimation versus the interacting multiple model algorithm," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-25, no. 2, pp. 296-300, Mar. 1989.
- [50] E. Emre and J. Seo, "A unifying approach to multitarget tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-25, no. 4, pp. 520-527, July 1989.
- [51] R. L. Kirlin and A. Moghaddamjoo, "Robust adaptive Kalman filtering for systems with unknown step inputs and non-Gaussian measurement errors," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 34, no. 2, pp. 252-263, Apr. 1986.
- [52] A. Moghaddamjoo and R. L. Kirlin, "Robust adaptive Kalman filtering with unknown inputs," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, no. 8, pp. 1166-1175, Aug. 1989.
- [53] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, 3rd ed. New York: McGraw-Hill, Inc., 1991.
- [54] J. W. Nilsson, *Electric Circuits*, 3rd ed. New York: Addison-Wesley, Inc., 1990.
- [55] J. K. Tugnait, "Comments on 'State estimation for discrete systems with switching parameters'," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-15, no. 3, p. 464, May 1979.
- [56] J. Raisch, "Comments on 'A multi-model adaptive predictor for stochastic processes with Markov switching parameters'," *Int. J. Control*, vol. 45, no. 4, pp. 1489-1490, 1987.
- [57] R. G. Brown, *Introduction to Random Signal Analysis and Kalman Filtering*, 2nd ed. New York: John Wiley and Sons, Inc., 1991.

- [58] J. S. Bendat and A. G. Piersol, *Engineering Applications of Correlation and Spectral Analysis*. New York: John Wiley and Sons, Inc., 1980.
- [59] H. F. VanLandingham, *Introduction to Digital Control Systems*. New York: Macmillan Publishing Co., 1985.
- [60] J. H. Mendel, *Lessons in Digital Estimation Theory*. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1987.
- [61] H. Stark and J. W. Woods, *Probability, Random Processes, and Estimation Theory for Engineers*. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1986.

A Computationally Efficient Technique for State Estimation Of Nonlinear Systems ¹

Jastej S. Dhingra
Richard L. Moose
Hugh VanLandingham and
Thomas A. Lauzon
of the

Bradley Department of Electrical Engineering
Virginia Polytechnic Institute and State University
Blacksburg, Virginia 24061

¹This work was supported by the Office of Naval Research under Contract N0014-89-J-3123

Key Words

Nonlinear estimation, nonlinear sampled-data simulation, computationally efficient estimation, state estimation.

Abstract

An estimation technique is presented for the class of nonlinear systems consisting of memoryless nonlinearities embedded in a dynamic linear system. The approach is based on a useful sampled-data non-linear system simulation method, which involves the addition of an extra state variable for each nonlinear element. The nonlinear estimator is developed along the lines of the basic Kalman state estimation, using quasi-linearization instead of the Taylor series linearization used in Extended Kalman Filters. It is demonstrated that this new method, out performs the extended Kalman filter in terms of the mean-square error of the state estimate. This estimator can be used effectively for state estimation for those cases where the extended Kalman filter doesnot converge. Moreover the new method is directly applicable to feedback systems with multiple nonlinearities and stochastic disturbances.

1. Introduction

Linear models are often used by engineers to simplify the analysis and design process for physical systems, but inevitably nonlinearities appear when the model is refined, and in many cases a linear model is too crude to be of use. Consequently, methods for working with nonlinear systems are always of great interest.

For on-line system identification and control it is customary to use an estimator to provide good state estimates from the available noise corrupted measurements. The past quarter century has witnessed many approaches to this problem, but most of the approaches resulted in filters with growing memory or of infinite dimension or which were simply cumbersome computationally. The extended Kalman filter (EKF) evolved as a general standard approach to this problem.

Although state estimation of linear systems has been thoroughly studied and documented, corresponding studies for nonlinear systems are still going on. In the majority of these studies attempts have been made to linearize or quasi-linearize the nonlinear model so as to apply the linear approaches. Ramnath and Paynter [1] used the idea of a scaling transformation to reduce the nonlinear problem to an obviously solvable form. For example, using the change of variables $z = \dot{y}/y$, the nonlinear Riccati equation:

$$\dot{z} + z^2 = 1$$

can be made equivalent to the *linear* equation:

$$\dot{q} - tq = 0$$

The main disadvantage of this method is that it is not generally clear how to construct an appropriate transformation.

Statistical linearization is another approach that has been proposed to deal with nonlinear systems. Craudall [2] explains that the idea is to replace the nonlinear

system equation by an equivalent linear equation involving a parameter which is selected to minimize the estimation error. The key step is the evaluation of statistical expectations of certain functions of the nonlinear response. But in any case this general approach only works well for small nonlinearities. Extensions of the statistical linearization technique were given by Hedrick and Arslan [3] and Beaman and Hedrick [1].

Yavin [5] discussed a modified version of the discrete Kalman filter for the nonlinear case. The essential idea is to have a nonlinear transition matrix A for the continuous case, develop the equivalent discrete transition matrix using the first two terms of the Taylor series expansion of $\exp(At)$ and complete the algorithm along the lines of linear Kalman filter.

Using the sampled-data technique presented by VanLandingham [6], VanLandingham and Moose[10] and Moose and Lauzon [7], Dhingra [8] later studied variations of estimators suggested by the Jump Matrix simulation method. The Jump Matrix modelling method has been successfully used to model large scale systems, such as the U.S. Navy *Close in Weapon System (CIWS)* containing as many as nine major non-linearities. Both the simulation technique and the resulting nonlinear estimator are discussed in detail in subsequent sections.

Section 2A presents the nonlinear sampled-data simulation method, followed by Section 2B in which the nonlinear estimation technique is developed. Section 3 gives some example simulations and comparisons with the extended Kalman filter.

2. A Sampled-Data Simulation Method for State Estimation of Nonlinear Stochastic Systems

A. Introduction

Starting with a basic state variable formulation for a plant model, a nonlinear sampled data simulation method called the Jump Matrix Technique (JMT) will be introduced. The nonlinear system simulation and design of the estimator using the JMT will thus form a major part of this section. The complete development of the technique is given in [7],[10].

Consider a general set of state equations describing a linear system

$$\dot{x} = Ax + Bu \quad \text{Plant Model} \quad (1)$$

$$y = Cx + Du \quad \text{Output Model} \quad (2)$$

If the system input is sampled at every T seconds and held constant over the time interval T , equation we have

$$x(t_{k+1}) = \Phi x(t_k) + \Gamma u(t_k) \quad (3)$$

$$y(t_k) = Cx(t_k) + Du(t_k) \quad (4)$$

where Φ and Γ are defined as follows

$$\Phi = e^{AT} = I + AT + A^2T^2/2! + \dots \quad (5)$$

$$\Gamma = \int_0^T e^{A\lambda} B d\lambda = (T + T^2A/2! + T^3A^2/3! + \dots)B. \quad (6)$$

Thus equation (3) defines the discrete time equivalent of the continuous time system given by (1a,b). In addition, these equations form the basis of system simulation for the linear time invariant system. Nonlinear system simulation is considered next.

The basis of the JMT is to split the nonlinear system into linear and nonlinear subsystems, using fictitious samplers and clamps. During the sample interval $t_k \leq t < t_{k+1}$ the linear subsystem propagates like an ordinary linear system. At the sampling instant $t = t_k$ the nonlinearity output is updated and then held constant over the sampling period. One implicit assumption in this analysis is that the nonlinearities are memoryless. Over the sampling period the clamping of the nonlinearity output

acts as an ideal integration of an input consisting of a series of impulses. The clamp output is then considered as an extra state variable of the system. Thus the nonlinear system is modeled as linear system during the sample interval with a nonlinear "jump" in the system state at the sampling instant. This is illustrated in Fig. 1.

For a small time interval, this becomes an accurate piece wise constant approximation of the actual nonlinearity. In terms of a set of state equations, the system using the JMT becomes

$$x(t_{k+1}^-) = \Phi x(t_k^+) + \Gamma u(t_k) \quad t_k < t < t_{k+1} \quad (7)$$

$$x(t_k^+) = J(x(t_k^-)) \quad t = t_k \quad (8)$$

Here $J(\cdot)$ is a nonlinear matrix operator, acting on the state vector $x(t_k^-)$ and will be referred to as the jump matrix or vector. At the sampling instant, only the nonlinear state variables are changed instantly; the linear variables are kept constant, which is in direct contrast with keeping the nonlinear state variable constant over the sampling period. As an example, consider the system defined by the equation

$$\ddot{x} + \dot{x}^2 x + x^2 \dot{x} = 0 \quad (9)$$

Choosing the two nonlinearities as state variables, the state equations can be set up as follows

$$\begin{aligned} \dot{x}_1 &= x \\ \dot{x}_2 &= \dot{x}_1 \\ \dot{x}_3 &= x_2^2 x_1 \\ \dot{x}_4 &= x_1^2 x_2 \end{aligned} \quad \begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -x_3 - x_4 \\ \dot{x}_3 &= 0 \\ \dot{x}_4 &= 0 \end{aligned} \quad t_k < t < t_{k+1}$$

Writing it in a matrix form

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} x \quad t_k < t < t_{k+1} \quad (10)$$

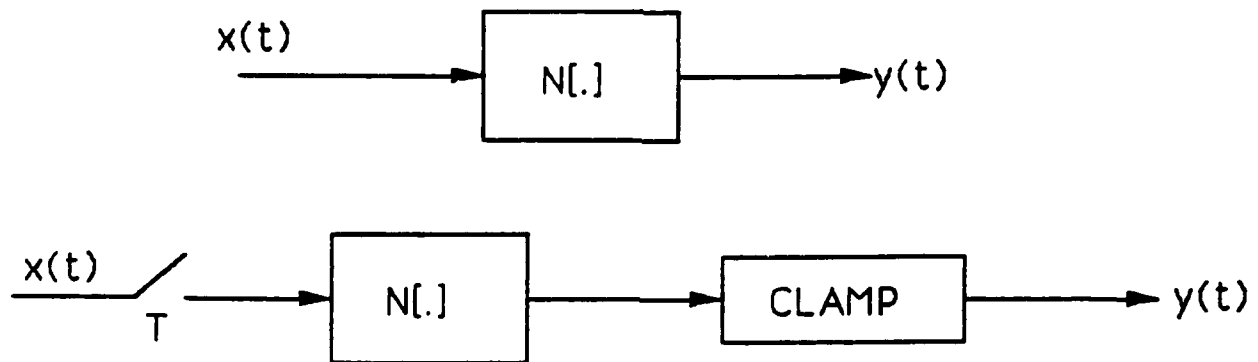


Figure 1: Induced Sample and Clamp for a Nonlinearity

where at $t = t_k$ the nonlinear (Jump) update is given by

$$x(t_k^+) = \begin{bmatrix} x_1(t_k^-) \\ x_2(t_k^-) \\ x_2^2(t_k^-), x_1(t_k^-) \\ x_1^2(t_k^-), x_2(t_k^-) \end{bmatrix} \triangleq J(x(t_k^-)), \quad t = t_k \quad (11)$$

B. Estimator Design

On the basis of the theory developed in Section 2A, the nonlinear stochastic system can be modeled as

$$\text{Nonlinear Jump:} \quad x_k^+ = J(x_k^-) \quad t = t_k$$

$$\text{Linear Projection:} \quad x_{k+1}^- = \Phi x_k^+ + \Gamma u_k + \Psi w_k \quad t_k < t < t_{k+1}$$

$$\text{Measurement:} \quad z_k^+ = H x_k^+ + v_k \quad t = t_k$$

The system noise w_k and measurement noise v_k are statistically independent, uncorrelated, gaussian random noises with covariance matrices Q and R respectively. Intuitively, the estimated state just after the sampling instant should be proportional to the nonlinear update of state before the sampling instant and also the measurement made at the sampling instant. So it is assumed that the filter has the following form:

$$\hat{x}_k^+ = \Theta J(\hat{x}_k^-) + \Delta z_k^+ \quad (12)$$

where Θ and Δ are weighting matrices that will be determined. During the duration of the sample interval T , the filter behaves as an ordinary linear estimator with the intersample response given by

$$\hat{x}_{k+1}^- = \Phi \hat{x}_k^+ + \Gamma u_k \quad (13)$$

Defining the filter error as $e_{k+1} = x_{k+1}^+ - \hat{x}_{k+1}^+$ and substituting the values from above we have

$$e_{k+1} = J(x_{k+1}^-) - \Theta J(\hat{x}_{k+1}^-) - \Delta [H J(\hat{x}_{k+1}^-) + v_{k+1}] \quad (14)$$

Taking the statistical expectation

$$E[\epsilon_{k+1}] = (I - \Delta H)E[J(x_{k+1}^-) - (I - \Delta H)^{-1}\Theta J(\hat{x}_{k+1}^-)] \quad (15)$$

For an unbiased estimator, we want $\hat{x}_k \rightarrow x_k$ as $k \rightarrow \infty$. So if $\Theta = (I - \Delta H)$ then as $k \rightarrow \infty$, $E[\epsilon_{k+1}] \rightarrow 0$. Thus equation (15) reduces to

$$E[\epsilon_{k+1}] = (I - \Delta H)E[J(x_{k+1}^-) - J(\hat{x}_{k+1}^-)] \quad (16)$$

The filter equation (12) can now be rewritten as

$$\hat{x}_k^+ = J(\hat{x}_k^-) - \Delta[\hat{z}_k - HJ(\hat{x}_k^-)] \quad (17)$$

which is similar to the state estimate update equation of the basic extended Kalman filter.

The problem now is to evaluate the statistical expectation of the term $[J(x_{k+1}^-) - J(\hat{x}_{k+1}^-)]$. Since $J(\cdot)$ is a nonlinear operator, it cannot be directly interchanged with expectation. Lauzon, Moose [7], used the idea of a time varying matrix a_{k+1} such that

$$[J(x_{k+1}^-) - J(\hat{x}_{k+1}^-)] = a_{k+1}[x_{k+1}^- - \hat{x}_{k+1}^-] \quad (18)$$

Determining a_{k+1} by direct manipulation of the definition is impossible as the value of the actual state is not known at each iteration. We consider the matrix a_{k+1} to be the *Jacobian matrix* of $J(\cdot)$ evaluated at some mean-value point \bar{x}_{k+1} lying in the closed interval $[x_{k+1}^-, \hat{x}_{k+1}^-]$. Again the interval is not defined without the knowledge of x_{k+1}^- . But since the estimator is unbiased, the interval gets smaller as $k \rightarrow \infty$, therefore the mean-value point is approximated by the estimated value \hat{x}_{k+1}^- . Proceeding with the estimator development, the expected value of the estimator error given by equation (16) becomes

$$E[\epsilon_{k+1}] = (I - \Delta H)a_{k+1}\Phi E[\epsilon_k] \quad (19)$$

and by making use of equation (14) and (18) the total error expression can be written as

$$\epsilon_{k+1} = (I - \Delta H)a_{k+1}(\Phi \epsilon_k + \Psi w_k) - \Delta v_{k+1} \quad (20)$$

The error covariance matrix P_{k+1} , defined as $E[\epsilon_{k+1}\epsilon_{k+1}^T]$, is determined using equation (20).

$$P_{k+1} = (I - \Delta H)[a_{k+1}(\Phi P_k \Phi^T + \Psi Q \Psi^T)a_{k+1}^T](I - H^T \Delta^T) + \Delta R \Delta^T \quad (21)$$

Defining

$$M_{k+1}^- = \Phi P_k \Phi^T + \Psi Q \Psi^T \quad (22)$$

and

$$M_{k+1}^+ = a_{k+1} M_{k+1}^- a_{k+1}^T \quad (23)$$

Equation (21) can be written as

$$P_{k+1} = (I - \Delta H)M_{k+1}^+(I - H^T \Delta^T) + \Delta R \Delta^T \quad (24)$$

In order to determine the optimal filter gain, the trace of the error covariance matrix P_{k+1} is minimized. Differentiating $C \triangleq \text{tr}(P_{k+1})$ with respect to Δ and setting the result equal to zero yields the optimal gain matrix

$$\Delta^* = M_{k+1}^+ H^T [H M_{k+1}^+ H^T + R]^{-1} \quad (25)$$

This expression looks very similar the Kalman gain equation. So defining Δ^* as the time varying gain K_{k+1} , P_{k+1} can be written in the recursive form as

$$P_{k+1} = (I - \Delta H)M_{k+1}^+(I - H^T \Delta^T) + \Delta R \Delta^T \quad (26)$$

which then reduces to the familiar expression

$$P_{k+1} = (I - K_{k+1} H)M_{k+1}^+ \quad (27)$$

This completes the design of the estimator. Summarizing the filter equations

$$\begin{aligned}
\text{Linear Prediction:} \quad \hat{x}_{k+1}^- &= \Phi \hat{x}_k^+ + \Gamma u_k \\
\text{Nonlinear Update:} \quad \hat{x}_{k+1}^+ &= J(\hat{x}_{k+1}^-) + K_{k+1}(z_{k+1}^+ - HJ(\hat{x}_{k+1}^-)) \\
\text{Gain Calculation:} \quad K_{k+1} &= M_{k+1}^+ H^T [H M_{k+1}^+ H^T + R]^{-1} \\
\text{Covariance Update:} \quad P_{k+1} &= (I - K_{k+1} H) M_{k+1}^+ \\
M_{k+1}^+ &= a_{k+1} \{ \Phi P_k \Phi^T + \Psi Q \Psi^T \} a_{k+1}^T \\
(a_{k+1})_{ij} &= \left. \frac{\partial f_i}{\partial x_j} \right|_{x=\hat{x}_{k+1}^-}
\end{aligned}$$

3. Estimator Analysis

A. Introduction

In section 2, the simulation of nonlinear systems using the Jump Matrix Technique (JMT) was first discussed and the technique used to develop an estimator for nonlinear systems. In this section two specific nonlinear systems are considered for the purpose of illustrating the estimator design. The first system chosen represents the motion of a particle under nonlinear damping. Van der Pol's equation can be used to model such a system. The second example considers a feedback control system containing an internal saturating nonlinearity. The system is influenced by the presence of wideband Gaussian disturbances and additive Gaussian measurement errors.

B. Van der Pol's Equation

The first example to be considered is the estimation of position of a particle or target under nonlinear damping as modeled by Van der Pol's equation. The solution of many nonlinear equations can be found by converting them into a form of Van der Pol's equation form, using time scaling, normalization, scaling transformation etc. A form of Van der Pol's equation driven by a gaussian wideband input $w(t)$ can be

written as

$$\ddot{x} - \epsilon(1 - x^2)\dot{x} + 9x = w(t) \quad (28)$$

Setting up the state equations

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \dot{x}_1 \\ \dot{x}_3 &= -\epsilon x_1^2 x_2 \end{aligned} \quad \begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -x_3 - x_1 \\ \dot{x}_3 &= 0 \end{aligned}$$

Writing it in a matrix form

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 \\ -9 & \epsilon & 1 \\ 0 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} w(t) \quad (29)$$

$$x(t_k^+) = J(x(t_k^-)) = \begin{bmatrix} x_1(t_k^-) \\ x_2(t_k^-) \\ -\epsilon x_1^2(t_k^-) x_2(t_k^-) \end{bmatrix} \quad (30)$$

The block diagram for the system is given in Fig. 2. The system was simulated for different values of ϵ i.e. 0.5, and 2.0. System simulation and state estimation was carried out for various values of σ_p^2 , the position measurement error variance, and σ_w^2 , the variance of $w(t)$. Table shows the total mean square error between the estimated and the real state values for different noise levels for the two estimators. Table 1 gives a comparison of the execution times for the two techniques. The numbers represent time in msec for 1000 iterations. The simulations were carried out on a SUN 3/110 machine using the Motorola 68881 floating point coprocessor.

For the various values of system noise power and measurement noise power the mean square error in the case of JMT Estimator was much smaller than for EKF. Though Table 2 suggests that EKF is efficient in terms of execution times, it was found that the times become comparable as the order of the system increases. The difference is basically due to the additional state variables introduced for the nonlinearities and hence the computations in case of JMT estimator are for a higher order system than the one used in the EKF case. But if performance, defined as the mean-square-error, is the criteria, the JMT estimator clearly out performs the EKF

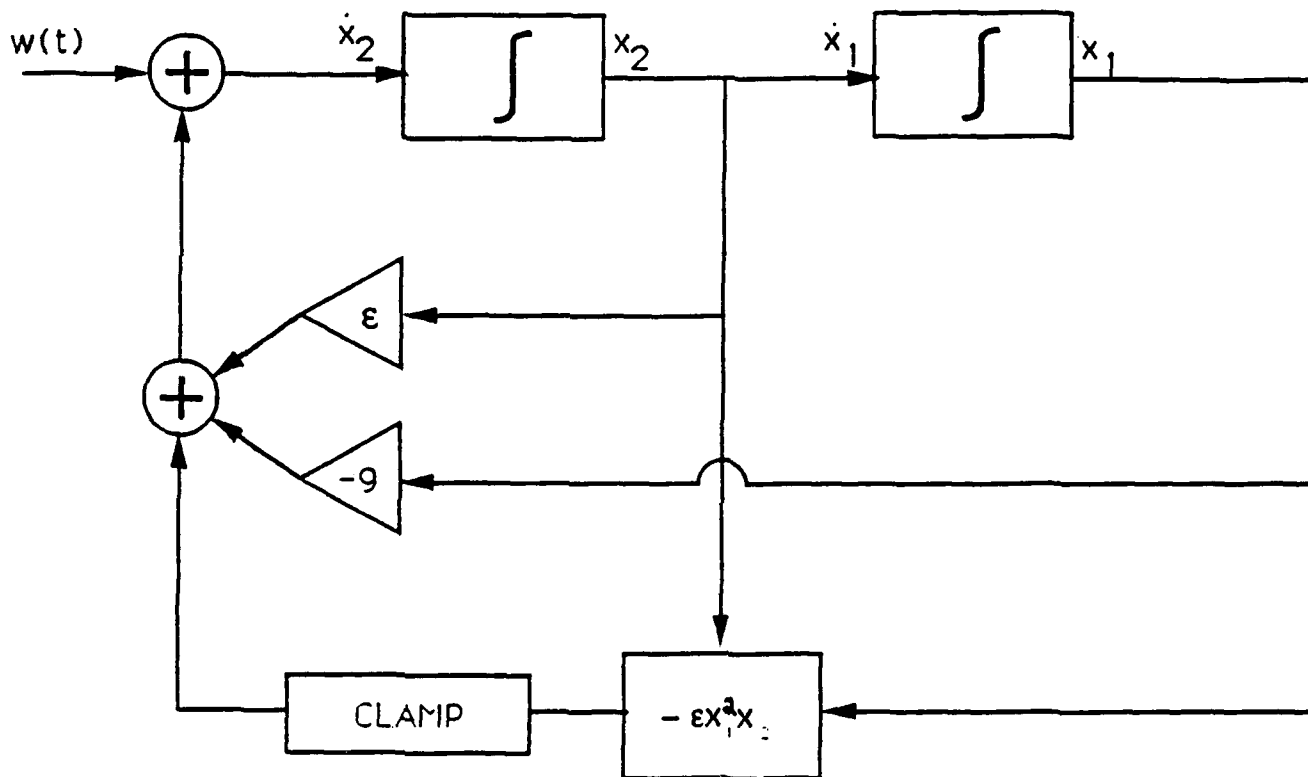


Figure 2: Block Diagram for Van der Pol's Equation

| | Extended Kalman Filter | JMT Estimator |
|--|---------------------------|---------------|
| $\sigma_v^2 = 0.01$ $\sigma_w^2 = 0.01$ | 0.1277 | 0.058293 |
| $\sigma_v^2 = 0.1$ $\sigma_w^2 = 0.01$ | 0.47089 | 0.34474 |
| $\sigma_v^2 = 0.1$ $\sigma_w^2 = 0.1$ | 0.67396 | 0.34524 |
| $\sigma_v^2 = 1.0$ $\sigma_w^2 = 0.1$ | 1.7192 | 1.336 |
| $\sigma_v^2 = 1.0$ $\sigma_w^2 = 1.0$ | 3.17196 | 1.34143 |

Table 1: Estimator mean square error. EKF vs JMT

| | Extended Kalman Filter | JMT Estimator |
|--|---------------------------|---------------|
| $\sigma_v^2 = 0.01$ $\sigma_w^2 = 0.01$ | 6.1 | 10.5 |
| $\sigma_v^2 = 0.1$ $\sigma_w^2 = 0.01$ | 5.8 | 10.1 |
| $\sigma_v^2 = 0.1$ $\sigma_w^2 = 0.1$ | 5.8 | 10.1 |
| $\sigma_v^2 = 1.0$ $\sigma_w^2 = 0.1$ | 5.9 | 10.2 |
| $\sigma_v^2 = 1.0$ $\sigma_w^2 = 1.0$ | 6.2 | 10.8 |

Table 2: Execution times on Sun 3/110. EKF vs JMT

and shows no tendency for filter divergence as is the case in EKF. The improvement in the estimator performance far outweighs the increase in execution time.

All the above simulations were carried out using $\epsilon = 2.0$. Fig. 3. shows the system state, the EKF estimate and the JMT estimate for the above system. $\epsilon = 2.0$, $\sigma_v^2 = 1.0$ and $\sigma_w^2 = 0.1$ were the set of parameters used for the figure. For the initial part both the estimated values are the same but gradually the JMT estimate tracks the input with smaller error and also with reducing time-lag. The actual system state variable values were computed using a different program, instead of the Jump simulation theory, and the same measurements were then used for both the estimators in order to have an unbiased comparison.

C. Non Linear Control System

The next example to be considered is a nonlinear Relay Control System. Kuo [9] has provided a discrete time analysis for calculation of the unit step response using phase plane trajectories. Here in this section a stochastic model of a nonlinear saturating amplifier is considered. The control system contains the amplifier followed by a second order Low Pass filter with negative output feedback. The block diagram for the system is given in Fig. 4. The system is modeled as follows

$$x_1 = y \quad (31)$$

$$x_2 = \dot{y} \quad (32)$$

$$x_3 = f(r - x_1 + w) \quad (33)$$

The state equations can now be set up in the matrix form as

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & 0 \end{bmatrix} x \quad (34)$$

with

$$x(t_k^+) = \begin{bmatrix} x_1(t_k^-) \\ x_2(t_k^-) \\ f(r(t_k) - x_1(t_k^-) + w_k) \end{bmatrix} = J(x(t_k^-)) \quad (35)$$

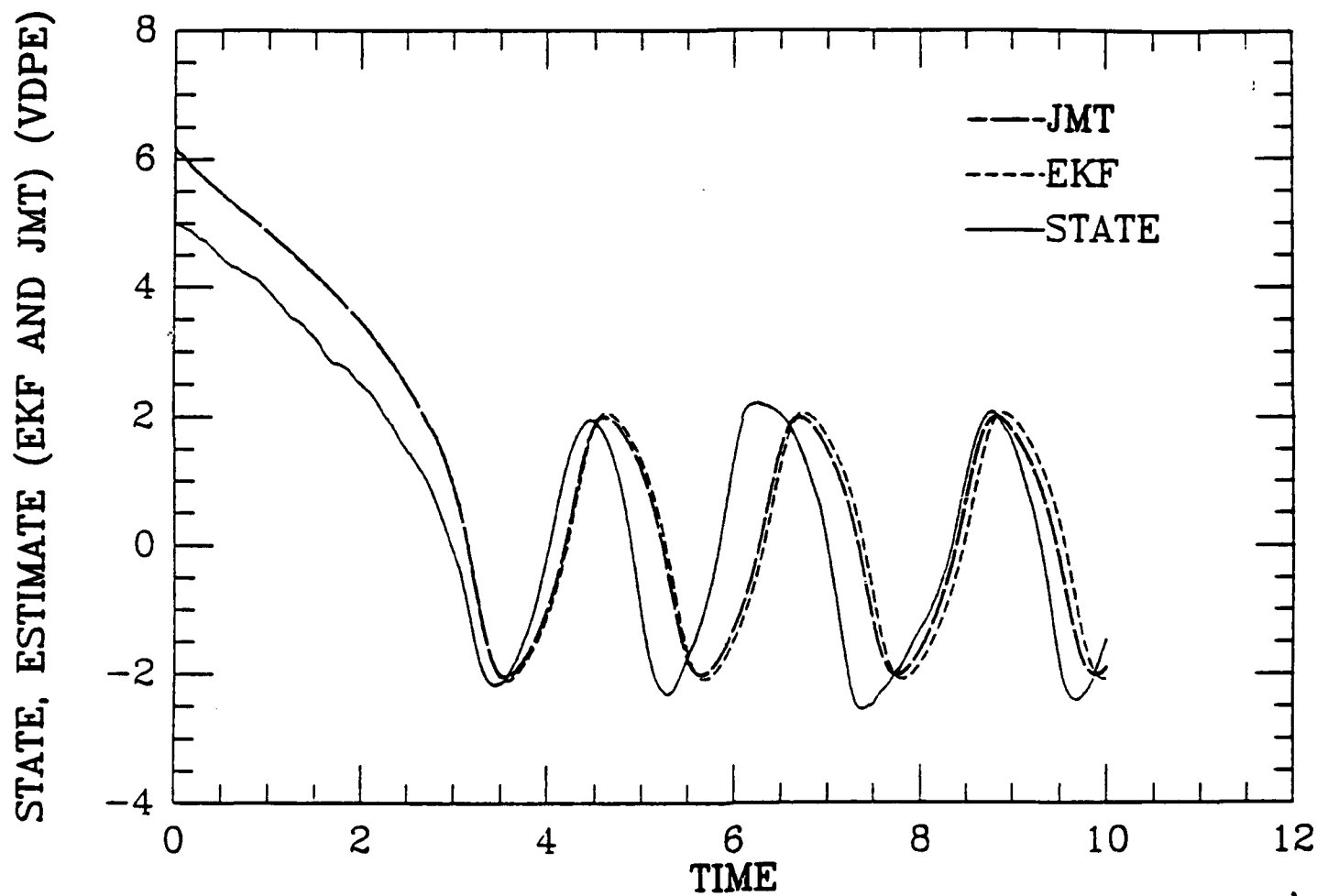


Figure 3: Van der Pol's equation. EKF vs JMT

The system simulation and estimation was carried out for $T=0.01$ sec, $\sigma_w^2 = 0.1$ and $\sigma_v^2 = 1$. The slope of the saturating amplifier was kept at 10.0. Fig. 5 shows the system state, EKF and JMT state estimates for the system. From the figure we observe that though the JMT estimate is noisy it still tracks the system state fairly well, while in the case of EKF the estimate *diverges* after some time. This further validates that the JMT estimator outperforms the EKF and also can be used for the cases where the EKF estimator fails.

5. Conclusion

A computationally efficient state estimator for nonlinear systems has been developed. The algorithm has been extensively tested and found to provide excellent state estimates for both open-loop and closed-loop systems with medium to low output signal-to noise ratios and stochastic or deterministic inputs.

The basic idea of the procedure involves a sampled-data approach and a separation of the linear and nonlinear parts of the system. During the sample interval, the linear dynamics propagate as expected, but the nonlinear component outputs are held constant until the end of the sample interval at which time they "jump" to an updated value. Consequently, the technique is referred to as the Jump Matrix Technique or JMT for short. The updating of the states at the sample instants introduces the effect of the nonlinear dynamics on the system. The estimator is developed on the basis of the predictor-corrector configuration.

The structure of the estimator algorithm includes a key time-varying gain matrix, a_{k+1} , which represents the relationship between the state estimation error before and after the nonlinear updates. In effect, a_{k+1} is a Jacobian matrix evaluated at the one-step predicted state estimate. This "quasi-linearization" of the nonlinear components is a frequently used method for approximating the nonlinear effects without undue computational demands on the algorithm. With the on-line calculation of the Jacobian matrix, the estimator takes on the form of a modified Kalman filter, but

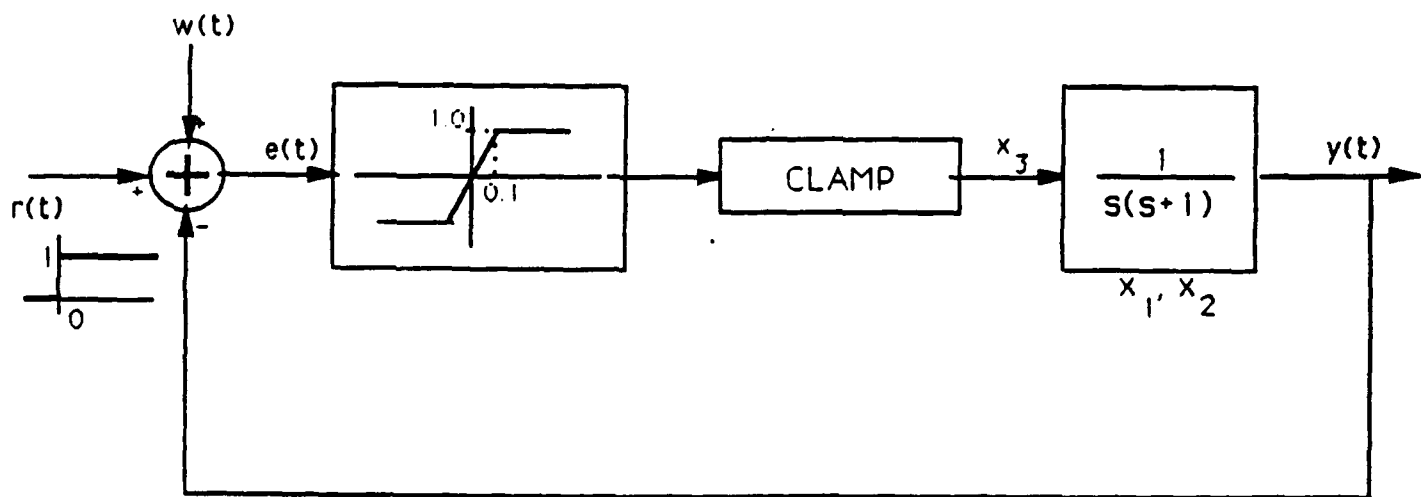


Figure 4: Block diagram of the control system with a saturating amplifier

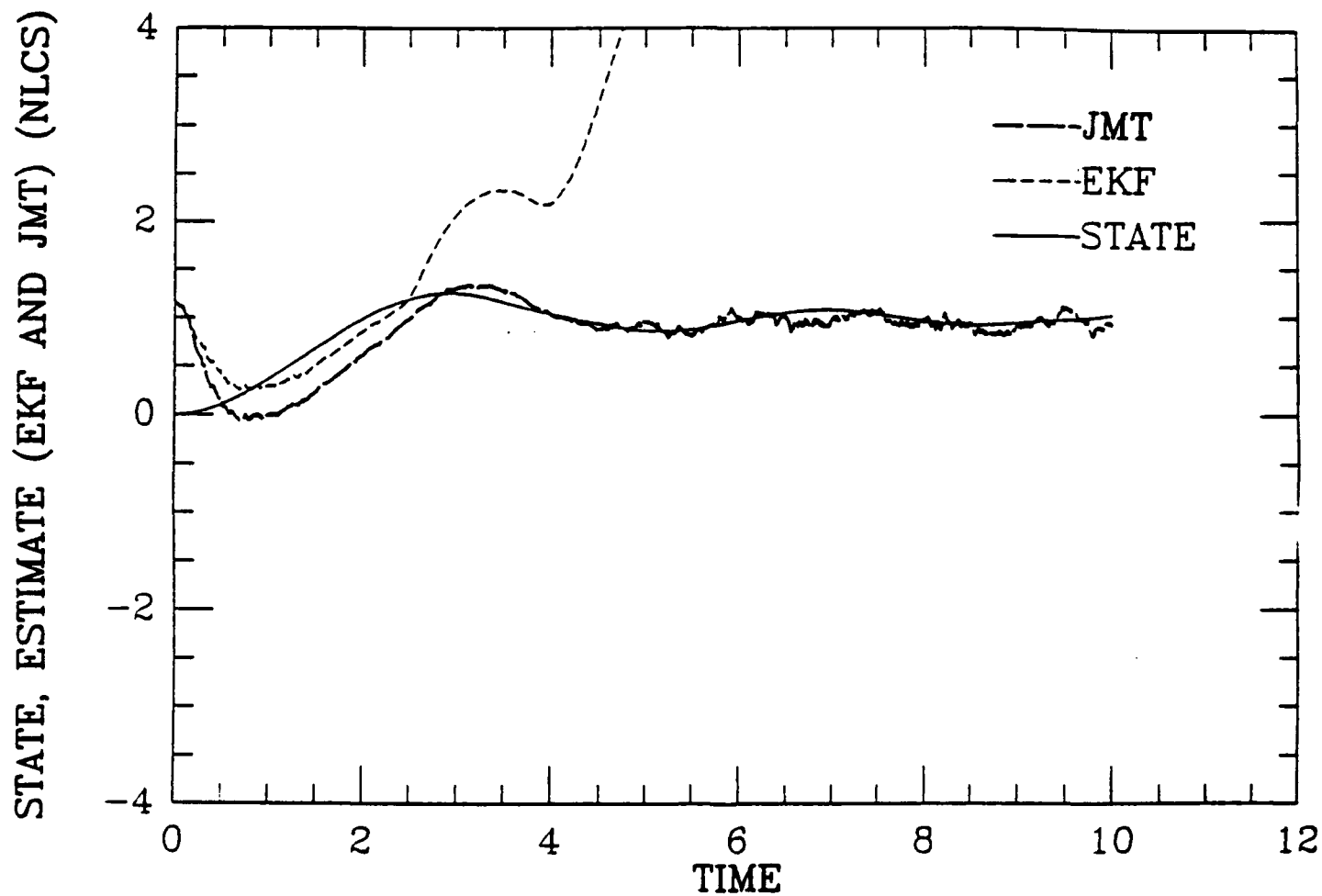


Figure 5: Non Linear Control System. EKF vs JMT

is distinct from and is shown to be superior to the Extended Kalman filter.

Two examples of stochastic nonlinear systems have been presented. The first was Van der Pol's equation illustrating nonlinear oscillation. In this case the mean square error of the estimate was consistently smaller for the JMT estimator, in comparison to the EKF. The second was a nonlinear feedback control loop with an internal saturating gain block, an example of systems extremely difficult to handle by Extended Kalman filtering techniques. The system was driven by a deterministic signal modeled by a unit step and a wideband noise disturbance $w(t)$. In addition, Gaussian measurement error $v(t)$ was introduced to yield low output signal to noise ratios. In this example the EKF state estimate diverged from the actual state, while the JMT estimator provided a close estimate of the system state. The JMT estimator performed very well in both examples, and in both cases gave better results than the EKF.

6. References

- [1]. Ramnath, R.V., Paynter, H.M., "Scaling Transformations in Nonlinear Systems", *Winter Annual Meeting of the American Society of Mechanical Engineers*, New York, NY, Dec. 1979.
- [2]. Crandall, S.H., "On statistical Linearization for Nonlinear Oscillators", *Winter Annual Meeting of the American Society of Mechanical Engineers*, New York, NY, Dec. 1979.
- [3]. Hedrick, J.K., Arslan, A.V., "Nonlinear analysis of Rail Vehicle Forced Lateral Response and Stability Analysis", *Winter Annual Meeting of the American Society of Mechanical Engineers*, New York, NY, Dec. 1979.
- [4]. Beaman, J.J., Hedrick J.K., "An Extended Statistical Linearization Technique" *Winter Annual Meeting of the American Society of Mechanical Engineers*, New York, NY, Dec. 1979.
- [5]. Yavin, Y., "Numerical studies in Nonlinear filtering", Springer Verlag, New

York, 1985.

[6]. VanLandingham, H.F., "Introduction to Digital Control Systems", Macmillan

- Publishing Company, New York, NY, 1985.

[7]. Lauzon, T.A., Moose, R.L., "An Application of digital simulation of Nonlinear estimation", Project Report, Virginia Polytechnic Institute and State University, Blacksburg, Virginia.

[8]. Dhingra, J.S., "Estimation of Nonlinear Systems Using Jump Matrix Technique", M.S. Thesis, EE Department VPI and State University, 1988.

[9]. Kuo, B.C., "Discrete Data Control Systems", Prentice Hall, Inc. Englewood Cliffs, NJ, 1970.

[10]. VanLandingham, H.F., Moose, R.L., "Digital Simulation of Nonlinear Stochastic Systems", In *Numerical Methods for Differential Equations and Simulation*, North-Holland Publishing Company, 1978, pp 207-213.

**NONLINEAR EFFECTS OF A MODAL DOMAIN OPTICAL FIBER SENSOR IN
A VIBRATION SUPPRESSION CONTROL LOOP FOR A FLEXIBLE
STRUCTURE**

Douglas K. Lindner, Gregory A. Zvonar, William T. Baumann,
and Peter L. Delos

The Bradley Department of Electrical Engineering
Virginia Tech
Blacksburg, Virginia 24061
(703) 231-4580
lindner@vtvm1.cc.vt.edu

revised
April, 1992

Accepted For
~~Submitted to~~
ASME Journal of Vibration and Acoustics

ABSTRACT

Recently, a modal domain optical fiber sensor has been demonstrated as a sensor in a control system for vibration suppression of a flexible cantilevered beam. This sensor responds to strain through a mechanical attachment to the structure. Because this sensor is of the interferometric type, the output of the sensor has a sinusoidal nonlinearity. For small levels of strain, the sensor can be operated in its linear region. For large levels of strain, the detection electronics can be configured to count fringes. In both of these configurations, the sensor nonlinearity imposes some restrictions on the performance of the control system. In this paper we investigate the effects of these sensor nonlinearities on the control system, and identify the region of linear operation in terms of the optical fiber sensor parameters.

1. INTRODUCTION

In this paper we are concerned with the design of active control systems for vibration suppression that contain a modal domain optical fiber sensor. A modal domain optical fiber sensor consists of a coherent light source (a laser), an optical fiber which responds to a measurand, and a detector. By knowing the relationship between the force on the structure and the output of the optical sensor, this sensor can be incorporated into a vibration suppression control system for a flexible beam (Cox and Lindner, 1991). In order to use such a sensor in a control system, Cox and Lindner (1991) developed an appropriate model for control system design that incorporated the following elements. When the optical fiber is attached to a flexible structure, force applied to the structure will induce a strain in the optical fiber (Mathews and Sirkis, 1991). (The relationship between the stress distribution in the host material and the optical fiber is currently an active area of research.) When the optical waveguide is subjected to strain, the intensity of the light at the fiber endface changes in a predictable way. The first model of this effect was reported by Butter and Hocker (1978). This model was developed further by Sirkis and Haslach (1991). This later work was extended slightly for modal domain sensors by Reichard and Lindner (1991). Last, the detection electronics (Murphy, et al., (1991)) are included in the model.

Modal domain optical fiber sensors are of interest for control system design because they can be configured to have a long gauge length. Recently, this class of sensors, called spatial filters, have been shown to have certain advantages when implementing complex control systems (Lindner, et al., 1990). They can also be configured for optimal measurement of vibrations in control systems for suppression of acoustic radiation (Lindner, et al., 1991a, 1991b). Reichard (1991) developed models to characterize fabrication and modeling errors for modal domain sensor when they are configured as spatial filters. Modal domain sensors have also been demonstrated in a control system for the suppression of acoustic radiation (Clark, et al, 1992).

Optical fiber sensors have certain advantages for instrumenting structural control systems. They are low power, light weight, low mass and EMI insensitive. These sensor can also be attached to or embedded in a structure.¹

The model of the sensor from strain in the fiber to intensity at the endface of the fiber (sensor output) contains a sinusoidal nonlinearity. Most of the results reported to date operate this sensor in a range of strain for which the output is in its linear range. In particular, the analysis, design, and experimental verification by Cox and Lindner (1991) of the vibration suppression control system for a cantilevered beam emphasized the linear

¹ See, for example the SPIE Proceedings on Fiber Optic Smart Structures and Skins, I-IV, 1988-1991.

region of operation of the sensor. In this paper we extend the analysis of the performance of the control system into the nonlinear region of the modal domain optical fiber sensor. In addition, we consider an alternative detection scheme which extends the dynamic range of the sensor. For both configurations, we investigate the existence of equilibrium points and limit cycles through a parametric study. We also characterize the distortion at the output of the sensor introduced by the sensor nonlinearities. Using these results, a modal domain optical fiber sensor can be sized for a particular application based on the predicted disturbance levels.

In Section 2 we introduce the model of the sensor and quantify the open loop distortion introduced by the sine nonlinearity. In Section 3 the effects of the nonlinearities on the closed loop system including the stability of the additional equilibrium points and limit cycles are investigated. Section 4 has the conclusions.

List of Symbols

Optical fiber parameters

- a - core radius of the optical fiber
- n_1 - index of refraction of the core
- n_2 - index of refraction of the cladding
- λ - wavelength of the laser
- I_f - intensity of the light at the fiber endface
- r, θ, z - cylindrical coordinate system for the optical fiber
- z_f - location of the endface of the fiber
- β_i - propagation constant of the i th electromagnetic mode
- Γ - phase of the interference pattern in the optical fiber
- $\tilde{\Gamma}_1$ - first order approximation of the phase to strain induced by stress
- $\tilde{\Gamma}_0(\epsilon^*)$ - Q-point of the sensor
- $\Delta\beta$ - fringe frequency,
- L_f - fringe length, period associated with the fringe frequency
- S - path of the optical fiber attachment to the structure
- $y_{MD}(t)$ - output of the sensor in its analog configuration
- $y_{FC}(t)$ - output of the sensor in its fringe counting configuration
- $Q_{FC}(\mu)$ - Quantizer nonlinearity
- $\Delta\Gamma_0$ - Q-point drift of the sensor
- J_i - i th order Bessel function of the first kind
- P_i - power in the i th harmonic
- THD - total harmonic distortion

Structure's parameters

$\epsilon_i(z,t)$ - strain tensor in the fiber at position z at time t
 σ_z - axial stress in the fiber
 $\epsilon(\sigma_z) = \epsilon_0$ - strain resulting from a unidirectional stress
 ϵ_0^* - prestrain in the fiber to set the Q-point

State space models

η - vector of the first N modal amplitudes
 η_e, x_{ce} - equilibrium points of the closed loop system
 ψ_i - i th mode shape
 ω_i - i th natural frequency of the structure
 Ω - diagonal matrix of the first N natural frequencies of the structure
 ζ_i - damping of the i th vibrational mode
 D - diagonal matrix of the damping factors of the modes of the structure
 c_i - "mode shape" resulting from the sensor placement
 B - matrix of modal influence coefficients
 x_p - state vector associated with the structure
 $(v(t), x(t), y(t))$ - input, state and output, respectively, of the dynamic compensator
 (A_c, B_c, C_c, J_c) - state matrices of the dynamic compensator
 k_d - closed loop gain of the control system
 k_l - closed loop gain of the control system which results in poles at $\pm j\omega_i$
 k_{dmax} - maximum closed loop gain of the control system such that the system will not have a limit cycle of frequency approximately ω_i
 k_c - steady state gain of the closed loop control system
 $P(s)$ - transfer function of the structure
 $C(s)$ - transfer function of the compensator
 $N(A)$ - describing function of a sinusoidal nonlinearity
 $N_{FQ}(A)$ - describing function of a quantizer nonlinearity
 A_0 - amplitude of a limit cycle

2. MODEL OF THE OPTICAL SENSOR

2.1 Introduction

The results reported in this paper were motivated by the experiment shown in Figure 2.1 (Cox and Lindner, 1991).

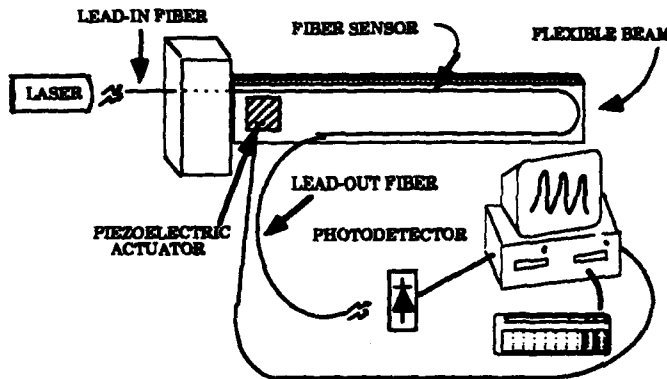


Figure 2.1. Flexible Beam with a Modal Domain Optical Fiber Sensor.

This experiment consisted of a flexible cantilevered beam with a piezoelectric bending motor attached at the root of the beam and a modal domain optical fiber sensor attached along the length of the beam. The output of the optical fiber was used as a feedback signal to damp vibrations in the beam. To describe the effects of the modal domain sensor on the performance of the control system, we require a model of the sensor. In this section we develop that model and show the effect of the nonlinearity on the sensor output.

2.2 Sensor Model

2.2.1 Introduction

A modal domain optical fiber sensor, shown in Figure 2.2 as a block diagram, consists of: 1) polarized laser light source, 2) lead-in optical fiber, 3) fiber sensing section, 4) lead-out fiber, and 5) detection electronics. This particular optical sensor is distinguished from other optical fiber sensors in that the waveguide parameters and source wavelength are chosen such that two electromagnetic modes propagate in the optical fiber. A modal domain optical fiber sensor was first demonstrated by Layton and Buccaro (1979). Elliptical-core fibers were introduced by Kim et al (1987) to stabilize the intensity pattern at the fiber interface. The first use of e-core fibers as strain gauges was reported by Blake, et. al (1987). Murphy, et al (1990) introduced lead-in and lead out fibers to isolate the gauge length of the sensor and discussed analog and fringe counting configurations of the detection electronics.

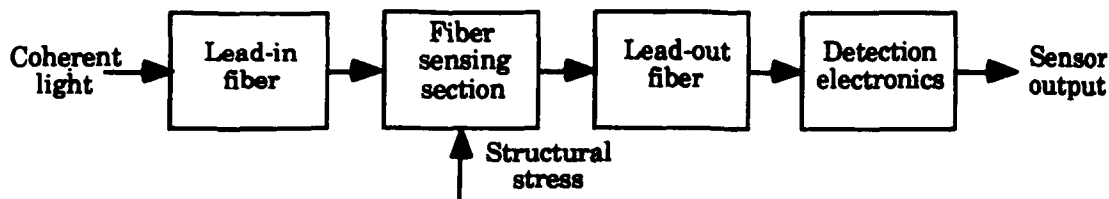


Figure 2.2. Block Diagram of the Modal Domain Optical Fiber Sensor.

The optical fiber sensor in Figure 2.1 is mechanically attached to the structure through an adhesive. (Optical fiber sensors can also be embedded in a composite.) When an external load is applied to the structure, stress is transferred from the material to the optical fiber. The stress in the optical fiber induces strain in the fiber. The change in strain in the optical fiber is observed as a change in the intensity at the endface of the fiber. Based on this observation, the model of the modal domain optical fiber sensor that can be described in terms of the following components:

1. The optical interrogation of the fiber.
2. The transfer of stress from the material to the optical fiber.
3. The strain-optic interaction.
4. The detection electronics.

In the analysis below we make the following assumptions for each of the components of the model above. The coordinate system of the optical waveguide is shown in Figure 2.3

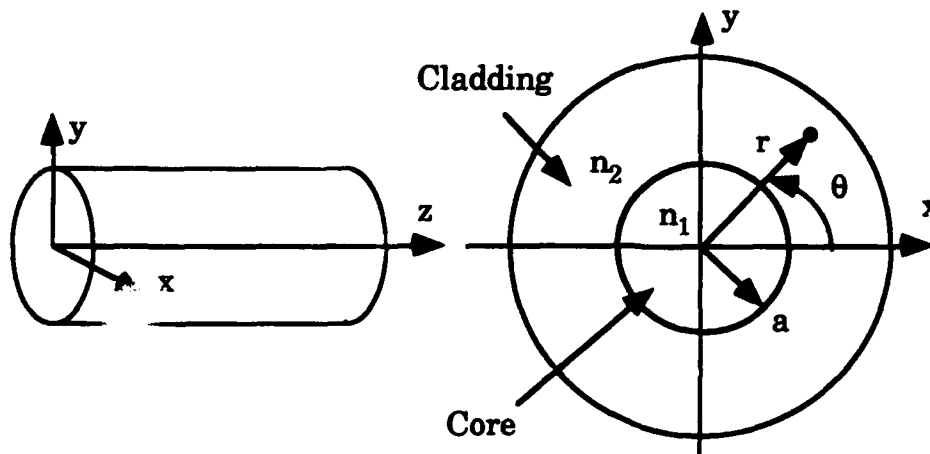


Figure 2.3. Coordinate System for the Optical Fiber.

1. The optical fiber are strands of glass configured to guide coherent light.
 - (a) We assume that the optical waveguide has a cylindrical geometry. This geometry is parameterized by the core radius, a .

- (b) We assume that the cladding is infinitely thick so that the weakly guiding assumption ($n_1 - n_2 \ll 1$) holds (Synder and Love, 1983). These assumptions are commonly made in the analysis of optical fibers. Using these assumptions for the waveguide, the electromagnetic modes are called LP modes.

The implementations of modal domain sensors use elliptical core fibers. The analysis of the electromagnetic modes that propagate in these waveguides is a currently topic of research.

2. A modal domain sensor measures strain in the structure through the mechanical attachment of the fiber to structure.
 - (a) We assume the stress in the host material is the same as the stress in the optical fiber. In real applications the model of the stress transfer between the host material and the optical fiber depends on several factors including the geometric orientation of the fiber on the structure and the properties of the bonding layer between the structure and the optical fiber.
 - (b) We assume that the strain in the fiber is uniform and that the change in the cylindrical geometry is such that the waveguide assumptions above are still valid.
3. The strain optic interaction obviously involves the assumptions in (1) and (2) above as well as the following assumption.
 - (a) We assume that there is no dynamic interaction between the stress waves in the glass of the optical fiber and the electromagnetic modes propagating in the waveguide.
4. A change in strain in the optical fiber results in a change in the intensity at the endface of the optical fiber.
 - (a) We assume that the electrical signal at the output of the detection electronics is proportional to the intensity of the light to a point at the endface of the fiber. The detection electronics for a modal domain sensor actually integrates the intensity of part of the endface of the fiber to increase the power coupling between the photodetector and the light. This configuration of the electronics does not impact the analysis below.

2.2.2 Optical Interrogation Of The Fiber

When the optical fiber is in an unstrained state, the fiber acts as a waveguide for the light. The guided light can be described in terms of eigensolutions of the governing partial differential equation by using

separation of variables. Each solution, an electromagnetic mode, that propagates in the waveguide depends on:

1. The geometry of the waveguide.
2. The indices of refraction of the waveguide n_1 and n_2 .
3. The laser source wavelength, λ .

For modal domain optical fiber sensors all of these parameters are chosen such that two electromagnetic modes propagate in the waveguide.

Using the weakly guiding assumption, the spatial distribution of the electric field of the two electromagnetic modes that propagate in the waveguide are of the form

$$E(r, \theta, z) = E_i(r, \theta)e^{-j\beta_i z}, \quad i = 0, 1, \quad (2.2.1)$$

where the constants $\beta_0 \neq \beta_1$ are the propagation constants. The propagation constants depend on the waveguide parameters above. These modes in (2.2.1) interfere with each other to produce a intensity pattern that varies along the length of the fiber. As a result of this interference, the intensity, I_f , at the endface of the fiber, z_f , has the functional form

$$I_f(r, \theta, z_f) = I_1(r, \theta) + I_2(r, \theta) \cos(\Delta\beta z_f + \alpha), \quad \Delta\beta = \beta_1 - \beta_0. \quad (2.2.2)$$

2.2.3 Strain Transfer

To predict the change in the sensor output with respect to a force applied to the structure requires a model of the strain transfer from the structure to the optical fiber. Our analysis below will assume that the functional relationship between the applied force from the actuator and the strain in the optical fiber is known. This area is topic of current research.

2.2.4 Strain Optic Interaction

Let the optical fiber be attached to the structure along a path S on the structure. Let $\epsilon(s, t)$ be the strain tensor for the point $s \in S$ at time t in the cylindrical coordinate system of the optical fiber. Let ϵ_1 be the normal component of the strain tensor aligned with the longitudinal axis of the fiber. The strain in the optical waveguide causes three of the parameters of the waveguide change:

1. The length of the optical fiber is changed; i.e. $z_f = z_f(\epsilon)$.

2. The core radius, a , is changed.

3. The indices of refraction, n_i , $i = 1, 2$, are changed through the photoelastic effect.

The dependence of these three parameters on strain in the optical fiber is reflected in the intensity of the light at the fiber interface (2.2.2) as

$$I_f(\epsilon) = I_1(\epsilon) + I_2(\epsilon) \cos(\Gamma(\epsilon)) \quad (2.2.3)$$

where

$$\Gamma(\epsilon) = \int_S \Delta\beta(\epsilon)(1+\epsilon_1)ds. \quad (2.2.4)$$

The quantity $\Gamma(\epsilon)$ is called the phase of the electromagnetic modes. It can be shown (Reichard and Lindner, 1991) that for typical values of optical fiber parameters the first order effects of the strain in the optical fiber is on the phase. Henceforth, we assume that the intensity terms in (2.2.3), I_1 and I_2 are independent of strain.

The functional dependence of the propagation constants on strain is nonlinear. Let ϵ^* be the strain distribution in the optical fiber when the structure is in equilibrium. The analysis proceeds by expanding the integrand in (2.2.4) in a Taylor series about ϵ^* to yield

$$\begin{aligned} \Delta\beta(\epsilon)(1+\epsilon_1) &= \Delta\beta(\epsilon^*)(1+\epsilon_1^*) \\ &+ \Delta\beta(\epsilon^*)(1+\epsilon_1) + \sum_{j=1}^6 \frac{\partial \Delta\beta(\epsilon^*)}{\partial \epsilon_j} (1+\epsilon_1^*)(\epsilon_j - \epsilon_j^*) + \text{h.o.t.} \end{aligned} \quad (2.2.5)$$

Dropping the higher order terms, the phase (2.2.4) can be rewritten as

$$\begin{aligned} \tilde{\Gamma}(\epsilon) &= \int_S \Delta\beta(\epsilon)(1+\epsilon_1)ds = \int_S \Delta\beta(\epsilon^*)(1+\epsilon_1^*) ds \\ &+ \int_S \Delta\beta(\epsilon^*)(1+\epsilon_1) + \sum_{j=1}^6 \frac{\partial \Delta\beta(\epsilon^*)}{\partial \epsilon_j} (1+\epsilon_1^*)(\epsilon_j - \epsilon_j^*) ds \\ &= \tilde{\Gamma}_0(\epsilon^*) + \tilde{\Gamma}_1(\epsilon). \end{aligned} \quad (2.2.6)$$

Substituting (2.2.6) into (2.2.3) we obtain

$$I_f(\epsilon) = I_1 + I_2 \cos(\tilde{\Gamma}(\epsilon)) = I_1 + I_2 \cos(\tilde{\Gamma}_0(\epsilon^*) + \tilde{\Gamma}_1(\epsilon)). \quad (2.2.7)$$

The quantity $\tilde{\Gamma}_0(\epsilon^*)$ is called the Q-point. The Q-point is set by static deformation in a non-sensing section of the optical fiber so that (2.2.7) becomes

$$I_f(\epsilon) = I_1 + I_2 \cos(\tilde{\Gamma}_0(\epsilon^*) + \tilde{\Gamma}_1(\epsilon)) = I_1 + I_2 \sin(\tilde{\Gamma}_1(\epsilon)). \quad (2.2.8)$$

The model of the modal domain sensor in (2.2.8) depends on the strain tensor. The results below do not depend on this tensor and they can be explained more simply by assuming the strain tensor ϵ is induced by a stress distribution in the optical fiber that can be modeled by single stress component σ_0 , i.e.

$$\epsilon(\sigma_0(s)) = \epsilon_0(s). \quad (2.2.9)$$

Using (2.2.9) the phase in the last term in (2.2.8) can be written as

$$\tilde{\Gamma}_1(\epsilon_0) = \int_S \Delta\tilde{\beta} \epsilon_0(s) ds = \Delta\tilde{\beta} \int_S \epsilon_0(s) ds. \quad (2.2.10)$$

Sensor output can be written as

$$I_f(t) = I_1 + I_2 \sin(\Delta\tilde{\beta} \int_S \epsilon_0(s, t) ds) \quad (2.2.11)$$

by substituting (2.2.10) into (2.2.8). A graph of the intensity at the endface of the fiber vs. the applied stress is shown in Figure 2.4.

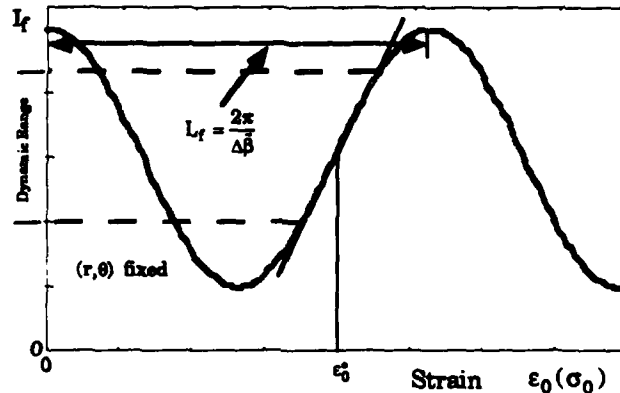


Figure 2.4. Intensity at the Fiber Endface vs. Strain.

From (2.2.11) we see that the sensor output has a sinusoidal nonlinearity. The frequency of this sine function, $\Delta\tilde{\beta}$, is called the fringe frequency. The period of this sine function,

$$L_f = \frac{2\pi}{\Delta\tilde{\beta}} \quad (2.2.12)$$

is called the fringe length.

2.2.5 Detection Electronics

The light at the endface of the fiber is converted into an electrical signal through the use of a photodetector. The output of the photodetector is taken to be proportional to the intensity of the light at a point of the endface of the fiber. The detection electronics use a highpass filter to remove the DC bias. The constant gain of the detection electronics is taken to be one for simplicity.

Combining all of the results above, the model of the modal domain sensor is

$$y_{MD}(t) = \sin\left(\Delta\tilde{\beta} \int_S \epsilon_0(s,t) ds\right). \quad (2.2.13)$$

When the modal domain sensor is used in its analog configuration, the analog signal from the detection electronics (2.2.13) is processed directly.

The second detection scheme counts the number of 2π phase shifts the output of the sensor experiences. We call this detection scheme fringe counting. To model this detector, we define the quantizer nonlinearity as

$$Q_{FC}(\mu) = nL_f, \quad \text{if } \left(n - \frac{1}{2}\right)L_f \leq \mu < \left(n + \frac{1}{2}\right)L_f. \quad (2.2.14)$$

The output of the detector is

$$y_{FC}(t) = Q_{FC}\left[\sin\left(\Delta\tilde{\beta} \int_S \epsilon_0(s,t) ds\right)\right]. \quad (2.2.15)$$

2.2.6 Sensor Parameters

The discussion above has identified the key parameters of a modal domain optical fiber sensor as far as the control system is concerned.

1. The most important parameter is the fringe frequency, $\Delta\tilde{\beta}$. For analog detection, this parameter essentially determines the dynamic range of the sensor as can be seen from Figure 2.4. For large amplitude strains, the output of this detector is dominated by the sine nonlinearity in (2.2.13). For fringe counting, the fringe frequency determines the quantization levels of the sensor output.
2. The second important parameter of this sensor is the Q-point. During the operation of the sensor, the Q-point can drift. If we let $\Delta\Gamma_0$ denote the Q-point drift the sensor output (2.2.13) is given by

$$y_{MD}(t) = \sin\left(\Delta\Gamma_0 + \Delta\tilde{\beta} \int_S \epsilon_0(s, t) ds\right). \quad (2.2.16)$$

In the sections below, the effect of Q-point drift is also discussed.

2.2.7 Scaling

Consider a structure which has been instrumented with a modal domain optical fiber sensor. If this structure is subjected to a sinusoidal force such that the sensor experiences a stress of, say,

$$\sigma_0(s, t) = F(s) \sin \omega_0 t, \quad (2.2.17)$$

then the mathematical model implies that the output of the sensor will be

$$\int_S \epsilon_0(s, t) ds = A \sin \omega_0 t \quad (2.2.18)$$

will also be sinusoidal with amplitude A, say. The relationship between $F(s)$ and A, which underlies the analysis presented here, depends on many factors including location of the force, material properties of the structure, frequency, etc. In addition, the sensor we consider here could be a long gauge length sensor. If this sensor is used in a long gauge length configuration, the effects of sensor placement should be considered on a case by case basis.

2.3. Large Amplitude Nonlinear Distortion

In this subsection we consider the modal domain sensor with analog detection electronics (2.2.13). For large levels of strain, however, the sine nonlinearity introduces distortion into the signal. In this section we quantify this distortion.

Suppose that a sinusoidal force on the structure results in a strain in the optical fiber that is given by (2.2.18). Substituting this expression for strain into the model of the sensor (2.2.11) we get

$$y_{MD}(t) = \sin\left(\Delta\Gamma_0 + \Delta\tilde{\beta} A \sin \omega_0 t\right). \quad (2.3.1)$$

Observing that the sensor output is a periodic function, (2.3.1) can be written as the Fourier series

$$\begin{aligned} y_{MD}(t) = & J_0(\Delta\tilde{\beta} A) \sin \Delta\Gamma_0 + \sum_{\substack{n=2 \\ n \text{ even}}}^{\infty} 2J_n(\Delta\tilde{\beta} A) \sin \Delta\Gamma_0 \cos n\omega_0 t \\ & + \sum_{\substack{n=1 \\ n \text{ odd}}}^{\infty} 2J_n(\Delta\tilde{\beta} A) \cos \Delta\Gamma_0 \sin n\omega_0 t \end{aligned} \quad (2.3.2)$$

where J_n is an n th order Bessel function of the first kind.

When $\Delta\tilde{\beta}A \ll 1$ and the Q-point drift is zero, $\Delta\Gamma_0 = 0$, the power in the signal $y_{MD}(t)$ is concentrated in the first term

$$y_{MD}(t) \approx 2J_1(\Delta\tilde{\beta}A) \sin \omega_0 t = \Delta\tilde{\beta}A \sin \omega_0 t. \quad (2.3.3)$$

With increasing amplitude of the strain, A , more power is shifted into the higher order harmonics in (2.3.2). Similarly, Q-point drift causes the appearance of a DC term as well as to shift power into the components $y_{MD}(t)$ which are out of phase with strain. This phenomenon can be quantified by defining the power in each harmonic as

$$P_i = \frac{(2J_n(\Delta\tilde{\beta}A) \sin \Delta\Gamma_0)^2}{2}, \quad n \text{ even}, \quad (2.3.4)$$

with a similar definition for n odd. Then the total harmonic distortion, THD, is defined as

$$THD = \sum_{i=2}^{\infty} \left[\frac{P_i}{P_1} \times 100\% \right]. \quad (2.3.5)$$

The THD is shown in Figure 2.5 as a function of the Q-point drift, $\Delta\Gamma_0$, the fringe frequency, $\Delta\tilde{\beta}$, and the amplitude of the strain, A . Figure 2.5 essentially describes the linear region of a modal domain sensor when it is used to sense vibrations in a flexible structure.

In a fringe counting configuration, this sensor will also introduce distortions in the sensor output because of the quantization. This effect has long been studied in the signal processing literature and it will not be pursued here.

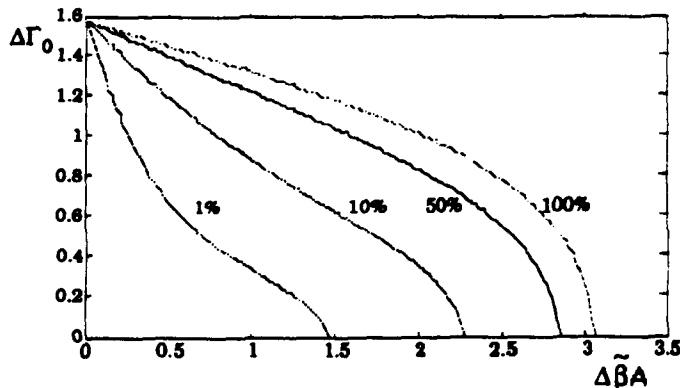


Figure 2.5. Total Harmonic Distortion.

CLOSED LOOP ANALYSIS

3.1 Introduction

In this section we consider the effect of the sine nonlinearity of the modal domain optical fiber sensor on the performance of a vibration suppression control system. To that end we assume that the structure is modeled by N vibrational modes as

$$\ddot{\eta}(t) + D\dot{\eta}(t) + \Omega^2\eta(t) = Bu(t) \quad (3.1.1)$$

here

$$\eta = \begin{bmatrix} \eta_1 \\ \vdots \\ \eta_N \end{bmatrix}, D = g(\zeta_1, \dots, \zeta_N), \Omega^2 = \text{diag}(\omega_1^2, \dots, \omega_N^2). \quad (3.1.2)$$

If we assume that the strain induced in the optical fiber by the vibrations of the structure can be expressed as a separation of variables expansion using the basis functions

$$\varepsilon_0(s, t) = \sum_{i=1}^N \psi_i(s) \eta_i(t), \quad (3.1.3)$$

then the sensor output is

$$\begin{aligned} y_{MD}(t) &= \sin \left[\Delta \tilde{\beta} \int_S \varepsilon_0(s, t) ds \right] = \sin \left[\Delta \tilde{\beta} \sum_{i=1}^N \eta_i(t) \int_S \psi_i(s) ds \right] \\ &= \sin \left[\Delta \tilde{\beta} \sum_{i=1}^N c_i \eta_i(t) \right] = \sin [\Delta \tilde{\beta} C \eta(t)]. \end{aligned} \quad (3.1.4)$$

The linearized model of the output (3.1.4) is

$$y(t) = \Delta \tilde{\beta} C \eta(t). \quad (3.1.5)$$

The linear open loop system, expressed in state space form, is

$$\begin{aligned} \dot{x}_p &= \frac{d}{dt} \begin{bmatrix} \eta(t) \\ \dot{\eta}(t) \end{bmatrix} = \begin{bmatrix} 0 & I \\ -\Omega^2 & -D \end{bmatrix} x_p + \begin{bmatrix} 0 \\ B \end{bmatrix} u(t), \\ y_{MD}(t) &= [\Delta \tilde{\beta} C \quad 0] x_p. \end{aligned} \quad (3.1.6)$$

The state space representation of the compensator is given by

$$\begin{aligned}\dot{\mathbf{x}}_c &= \mathbf{A}_c \mathbf{x}_c + \mathbf{B}_c v(t), \\ y_c &= \mathbf{C}_c \mathbf{x}_c + \mathbf{D}_c v(t).\end{aligned}\quad (3.1.7)$$

If the systems in (3.1.6) - (3.1.7) are interconnected according to

$$u(t) = y_c(t), \text{ and } v(t) = -k_d y_{MD}(t), \quad (3.1.8)$$

then the resulting state space representation for the closed loop system is

$$\begin{bmatrix} \dot{\mathbf{x}}_p \\ \dot{\mathbf{x}}_c \end{bmatrix} = \begin{bmatrix} 0 & \mathbf{I} & 0 \\ -\Omega^2 & -\mathbf{D} & \mathbf{B}\mathbf{C}_c \\ 0 & 0 & \mathbf{A}_c \end{bmatrix} \begin{bmatrix} \eta \\ \dot{\eta} \\ \mathbf{x}_c \end{bmatrix} - \begin{bmatrix} 0 \\ \mathbf{B}\mathbf{D}_c \\ \mathbf{B}_c \end{bmatrix} k_d \sin(\Delta\beta\mathbf{C}\eta). \quad (3.1.9)$$

The closed loop system is shown in Figure 3.1.

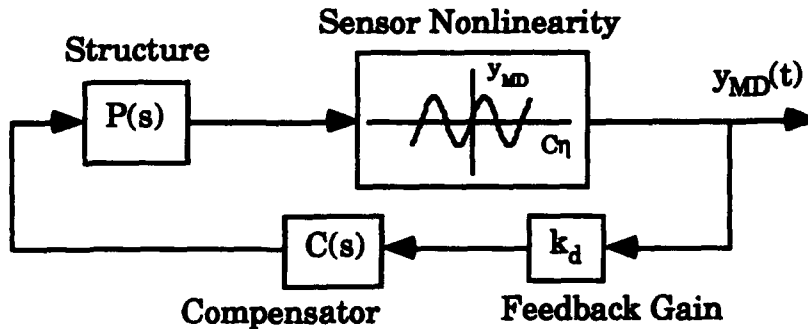


Figure 3.1. Closed Loop System.

In constructing the closed loop system (3.1.9) we only assumed that the compensator (3.1.7) gives acceptable closed loop performance with respect to the linear model (3.1.6). The actual design of the compensator could have been carried out on a reduced order model, say, a subset of the modes in (3.1.6). In that case the models in (3.1.6) and (3.1.9) would also include residual modes.

3.2 Equilibrium Points

The sine nonlinearity in the sensor introduces multiple equilibrium points into the linear design model. These equilibrium points are computed by setting

$$\dot{\mathbf{x}}_p = \dot{\mathbf{x}}_c = 0. \quad (3.2.1)$$

Substituting (3.2.1) into (3.1.9) we obtain

$$\begin{aligned}
0 &= \dot{\eta}, \\
0 &= -\Omega^2 \eta_e + B C_c x_{ce} - B D_c K_d \sin(\Delta \tilde{\beta} C \eta_e), \\
0 &= A_c x_{ce} - B_c k_d \sin(\Delta \tilde{\beta} C \eta_e).
\end{aligned} \tag{3.2.2}$$

Assuming A_c^{-1} exists, the last two equations in (3.2.2) can be rewritten as

$$\eta_e = -\Omega^{-2} B_c k_c k_d \sin(\Delta \tilde{\beta} C \eta_e), \tag{3.2.3}$$

where

$$k_c = D_c - C_c A_c^{-1} B_c. \tag{3.2.4}$$

Note that (3.2.4) is the steady state gain of the compensator.

The equilibrium points can be found by parameterizing (3.2.3) as

$$\eta_e = (\Omega^{-2} B_c k_c k_d) \gamma. \tag{3.2.5}$$

Using (3.2.5) in (3.2.3) we see that γ should satisfy

$$\gamma = -\sin(\Delta \tilde{\beta} C \Omega^{-2} B_c k_c k_d \gamma). \tag{3.2.6}$$

The scalar γ can be found graphically by plotting both sides of (3.2.6) on the same graph. A typical graphical solution to (3.2.6) is shown in Figure 3.2.

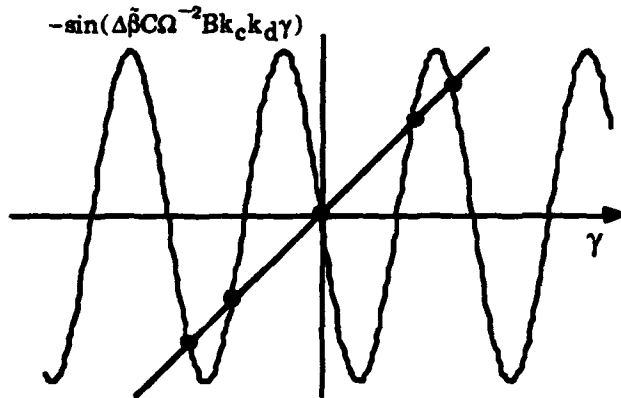


Figure 3.2. Calculation of Equilibrium Points.

A possible set of equilibrium points is also shown in Figure 3.2.

Based on the analysis above we can draw the following conclusions.

1. The assumption that Ω^{-2} exists implies that the model of the structure does not have any rigid body modes. Since modal domain sensors respond to strain this assumption is justified.
2. If A_c^{-1} does not exist, A_c has a zero eigenvalue corresponding to an integrator. In this case (3.2.2) may admit multiple solutions for x_{ce} .
3. If $k_c = 0$, the compensator has a zero at the origin which corresponds to pure velocity feedback. In this case the nonlinear system has only one equilibrium point at the origin.
4. Suppose that in the closed loop system (3.1.9), the structure is modeled by a single mode with a colocated force input and displacement output. Suppose further that the compensator is a simple constant $k_c = D_c > 0$ and $k_d > 0$. Then the sign convention we have chosen would result in increased stiffness in the closed loop system. If $k_d < 0$ the system would be unstable for large enough gains.
5. The number of equilibrium points is determined by the frequency of the sine function in (3.2.6). From Figure 3.2 we see that as the frequency of the sine function is increased, the number of equilibrium points is increased. The presence of multiple equilibrium points can be characterized in terms of three factors.
 - i) The factor $C\Omega^{-2}B$ is the steady state gain of the structure's transfer function. This factor includes the placement of the actuator and sensor through B and C matrices as well as the structure's modes.
 - ii) The second factor is $k_c k_d$. This factor represents the steady state gain supplied by the compensator. As this gain is increased, the number of equilibrium points increases as expected.
 - iii) The third factor is the fringe frequency, $\Delta\tilde{\beta}$, of the modal domain sensor. Increasing the fringe frequency decreases the dynamic range of the sensor, and increases the number equilibrium points.
6. The graph in Figure 3.2 assumed that the product of all of the factors in (3.2.6) was positive. If that product is negative, then the sine function is shifted by 180° . The results above remain essentially unchanged.
7. If the sensor has some Q-point drift, then (3.2.3) becomes

$$\eta_e = -\Omega^{-2}B_c k_c k_d \sin(\Delta\Gamma_0 + \Delta\tilde{\beta}C\eta_e). \quad (3.2.7)$$

The Q-point drift introduces a phase shift into the sine function in Figure 3.2 that causes all of the equilibrium points to shift. In particular, the equilibrium point at the origin is shifted to a nonzero value.

3.3 Stability of the Equilibrium Points

Stability Analysis

Next we use Lyapunov's first method to check the stability of the equilibrium points of the closed loop system (3.4.2). To this end we linearize (3.4.2) to obtain

$$\begin{bmatrix} \dot{x}_p \\ \dot{x}_c \end{bmatrix} = \begin{bmatrix} 0 & I & 0 \\ a_{21} & -D & BC_c \\ a_{31} & 0 & A_c \end{bmatrix} \begin{bmatrix} x_p \\ x_c \end{bmatrix} \quad (3.3.1)$$

where

$$\begin{aligned} a_{21} &= -\Omega^{-2} - \cos(\Delta\tilde{\beta}C\eta_e)k_d\Delta\tilde{\beta}BD_cC, \\ a_{31} &= -\cos(\Delta\tilde{\beta}C\eta_e)k_d\Delta\tilde{\beta}BC \end{aligned} \quad (3.3.2)$$

We call the factor

$$-k_d \cos(\Delta\tilde{\beta}C\eta_e) \quad (3.3.3)$$

the effective loop gain. For the equilibrium point $\eta_{e0} = 0$, the effective loop gain

$$-k_d \cos(\Delta\tilde{\beta}C\eta_{e0}) = -k_d \quad (3.3.4)$$

has its desired value. That is to say the poles of the system linearized around the equilibrium point at the origin correspond to the closed loop poles that resulted from the compensator designed using the linearized plant model. At nonzero equilibrium points, the effective loop gain has a value which is proportional to the slope of the term

$$-\sin(\Delta\tilde{\beta}C\Omega^{-2}Bk_c k_d \gamma) \quad (3.3.5)$$

at the intersections with the 45° line. See Figure 3.2.

We can draw the following conclusions:

1. We assume that the structure's poles are exactly on the imaginary axis (no natural damping), and that the poles depart into the left hand plane as the compensator gain is increased from 0 to k_d as shown in Figure 3.3. If the sign of the effective gain is reversed, then the poles of the linearized system in (3.1.5) will depart into the right half plane for small values of k_d and the corresponding equilibrium point are likely be unstable. Figure 3.2 shows that the equilibrium points are likely alternate between stability and instability for systems with no natural damping.
2. If the structure has some natural damping, the reasoning in (1) still holds qualitatively.
3. Suppose that the closed loop (3.1.9) system (3.3.1) has at least one nonzero equilibrium point corresponding, say, to the first intersection of the curves in Figure 3.2 for positive γ . At this intersection the slope of the sinusoid (3.3.5) is negative. Also suppose that the system has enough damping so that for some value of the fringe frequency, $\Delta\tilde{\beta}$, this equilibrium point is stable. As the fringe frequency increases, reducing the dynamic range, the slope of the intersection of the two curves in Figure 3.2 increases. As the fringe frequency increases the magnitude of the effective gain increases. As the fringe frequency increases to infinity, eventually this stable equilibrium point will become unstable.

Now consider the next equilibrium point of increasing γ in Figure 3.2. As the fringe frequency increases, the slope of the sinusoid increases and this equilibrium point remains stable. Increasing the fringe frequency decreases the stability region of each equilibrium point.

4. If the sensor has some Q-point drift, then the effective gain becomes

$$-k_d \cos(\Delta\Gamma_0 + \Delta\tilde{\beta}C\eta_e). \quad (3.3.6)$$

The Q-point drift causes the phase of the sinusoid in Figure 3.2 to shift. For Q-point drifts larger than 90° , an equilibrium point can change from stable to unstable.

3.4 Limit Cycles

Next we investigate the presence of limit cycles in a control system which incorporates a modal domain sensor. If the structure's transfer function is

$$P(s) = C(s^2I + Ds + \Omega^2)^{-1}B \quad (3.4.1)$$

and the compensators transfer function is

$$C(s) = [C_c(sI - A_c)^{-1}B_c + D_c], \quad (3.4.2)$$

then the closed loop system (3.1.9) is shown in the block diagram in Figure 3.1. We also assume that the closed loop system has only one equilibrium point at the origin.

To investigate the possible presence of limit cycles, we use describing functions (Atherton, 1975). We assume that the limit cycle at the input of the nonlinearity in Figure 3.1 has the form

$$A_0 \cos \omega_0 t, \quad (3.4.3)$$

and we look for conditions under which such a signal could be supported throughout the system. To that end, the describing function, $N(A)$, for the sinusoidal nonlinearity in Figure 3.1 is

$$N(A) = \frac{2J_1(\Delta\beta A)}{A}. \quad (3.4.4)$$

The graph of

$$\frac{N(A)}{\Delta\beta} = \frac{2J_1(\Delta\beta A)}{\Delta\beta A}. \quad (3.4.5)$$

is shown in Figure 3.3.

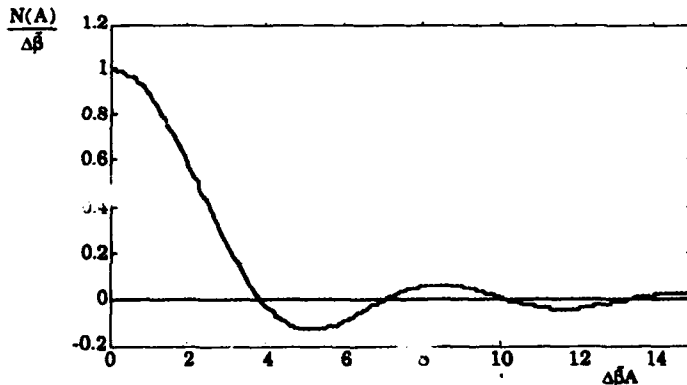


Figure 3.3. Describing Function of \tan^{-1} Sine Nonlinearity.

If this system has a limit cycle, then a solution, (A_0, ω_0) , to the equation

$$1 + N(A)k_d C(j\omega)P(j\omega) = 0 \quad (3.4.6)$$

should exist for some A_0 and ω_0 . Solutions to (3.4.6) can be found by plotting the root locus of

$$1 + k(A)C(s)P(s) = 0 \quad (3.4.7)$$

where

$$k(A) = N(A)k_d, \text{ for } 0 \leq A < \infty. \quad (3.4.8)$$

If the root locus intersects the imaginary axis at ω_0 for a value of $A = A_0$, then a limit cycle of the form (3.4.3) is predicted.

To investigate the presence of limit cycles for this system, we recast the root locus in (3.4.7) - (3.4.8) in terms of the root locus design of the original linear system as shown in Figure 3.4.

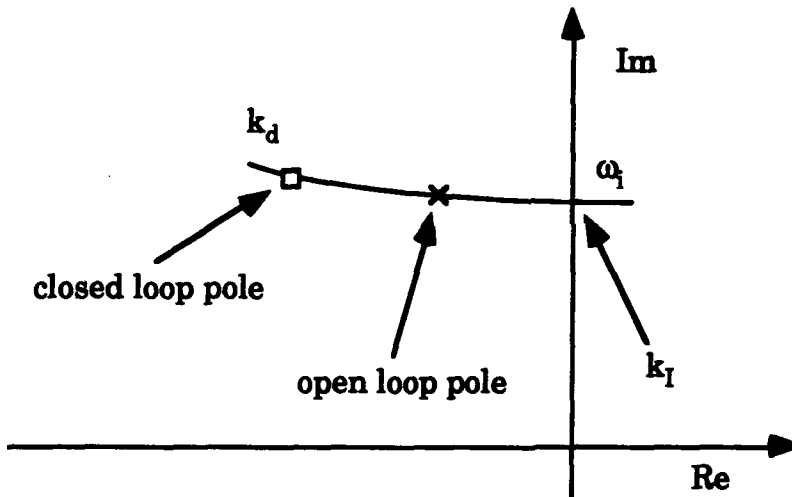


Figure 3.4. Root Locus of One Natural Frequency of the Structure.

Rewrite (3.1.7) as

$$v(t) = -kC\eta(t). \quad (3.4.9)$$

Then as k varies from $-\infty$ to ∞ , the poles of the linear system trace out the usual root locus curves. In particular, the closed loop poles are given by $k = k_d$.

From (3.4.8) we can see that for $A = 0$, $k = k_d \Delta \tilde{\beta}$. Let A_0 be the value of A that solves

$$J_1(\Delta\tilde{\beta}A) = 0. \quad (3.4.10)$$

Then this value of A corresponds to $k = 0$, i.e. the open loop poles of the structure. Note that larger values of A correspond to a sign change in the compensator gain k . As A goes to infinity, the gain in (3.4.8) oscillates between its minimum and maximum values, $-0.4\Delta\tilde{\beta}$, and $\Delta\tilde{\beta}$, respectively. Thus, as a function of A the variation of each pole of (3.4.7) is a single line beginning on the positive root locus and ending at the open loop pole which doubles back on itself. This behavior for the natural frequency ω_i of the structure is shown in Figure 3.5.

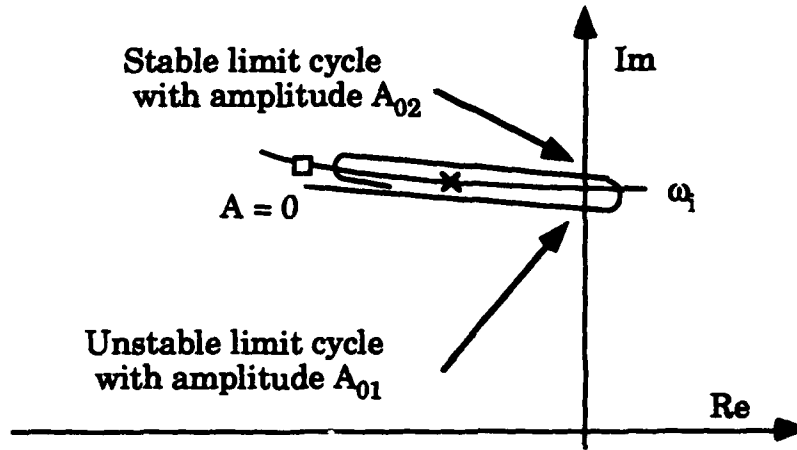


Figure 3.5. Root Locus Of the Describing Function of One Natural Frequency of the Structure.

Let k_I be the gain in (3.4.7) such that the pole at $j\omega_i$ is moved to the imaginary axis. Suppose that $k_I < 0$ as suggested in Figure 3.5. Next define A_{\min} as the constant that minimizes

$$\min_A N(\Delta\tilde{\beta}A). \quad (3.4.11)$$

Finally, define the feedback gain $k_{d\max}$ by

$$\frac{k_I}{\Delta\tilde{\beta}N(\Delta\tilde{\beta}A_{\min})} = k_{d\max}. \quad (3.4.14)$$

For a given structure, actuator, and modal domain sensor, if the gain of the closed loop system is

$$k_d > k_{d\max} \quad (3.4.15)$$

then the closed loop system will admit a limit cycle with a frequency of approximately ω_1 as shown in Figure 3.5.

Based on the analysis above, we can draw several conclusions.

1. For the example shown in Figure 3.5, if limit cycles exist, they are approximately at the same frequency as the natural frequency of the structure.
2. The value of the gain k_I , which determines k_{dmax} , will be a function of the material properties of the structure as well as the sensor and actuator placement.
3. If k_d , satisfies

$$0 < k_d < k_{dmax} \quad (3.4.16)$$

then this analysis does not predict any limit cycles. This range of gains depends on the fringe frequency as shown in (3.4.14). As the fringe frequency increases, k_{dmax} decreases as we would expect.

4. The amplitude of a limit cycle is determined from

$$k_I = N(\Delta\tilde{\beta}A_0)k_d \quad (3.4.17)$$

which may have multiple solutions for A_0 . For a given solution which satisfies

$$\Delta\tilde{\beta}A_0 = \text{constant} \quad (3.4.18)$$

the amplitude of the limit cycle A_0 will increase as the fringe frequency, $\Delta\tilde{\beta}$, decreases.

5. Suppose that a closed loop system admits the existence of multiple limit cycles with amplitudes, $A_{01} < \dots < A_{0(i+1)}$. It can be shown (Atherton, 1975) using standard arguments that the limit cycle corresponding to the amplitude A_{01} is unstable, and the limit cycle corresponding to A_{02} is stable. The limit cycles, ordered according to the magnitude of their amplitude, oscillate between stable and unstable, the first being unstable. This analysis is applied to each of the structure's natural frequencies.
6. If a closed loop system does admit at least one limit cycle, a rough measure of the linear operating region could be taken to be the region inside the limit cycle with smallest amplitude. For a given system, this region could be calculated from the analysis above.

7. This analysis focused on one gain for which the root loci crossed the imaginary axis. This analysis could be repeated at other gains if the root locus is more complicated than shown in Figure 3.5.

3.5 Fringe Counting

In the previous subsection we analyzed the effect of a modal domain optical fiber sensor in a feedback loop when it was in its analog configuration. In this subsection we consider a feedback loop with a modal domain sensor in its fringe counting configuration. In this configuration, the model of the sensor is a quantizer where the quantization levels are the fringe length L_f as shown in (2.2.14-15). The upper bound on the sensor output is determined by the material properties of the structure or the optical fiber; the strain level at which the material enters its plastic region. Another constraint is posed by the digital hardware. Here we assume that the upper bound is infinite.

The model of the closed loop system incorporating an modal domain sensor in a fringe counting configuration can be obtained from (3.1.9) by replacing the sine nonlinearity by the quantization nonlinearity (2.2.15). The result is

$$\begin{bmatrix} \dot{x}_p \\ \dot{x}_c \end{bmatrix} = \begin{bmatrix} 0 & I & 0 \\ -\Omega^{-2} & -D & BC_c \\ 0 & 0 & A_c \end{bmatrix} \begin{bmatrix} \eta \\ \dot{\eta} \\ x_c \end{bmatrix} - \begin{bmatrix} 0 \\ BD_c \\ B_c \end{bmatrix} k_d Q_{FC}(\Delta\Gamma_0 + \Delta\tilde{\beta}C\eta). \quad (3.5.1)$$

The equilibrium points satisfy

$$\begin{aligned} \eta_e &= (\Omega^{-2}Bk_c k_d)\gamma, \\ \gamma &= -Q_{FC}(\Delta\Gamma_0 + \Delta\tilde{\beta}C\Omega^{-2}Bk_c k_d\gamma) \end{aligned} \quad (3.5.2)$$

where the derivation of (3.5.2) follows the derivation of (3.2.5-6). Equation (3.5.2) can be solved graphically as shown in Figure 3.6.

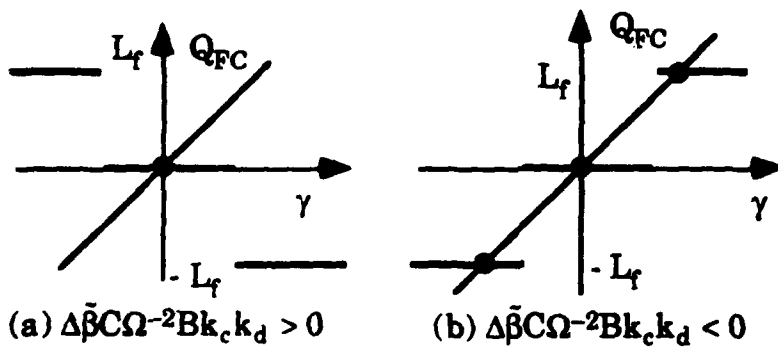


Figure 3.6. Equilibrium Points of the Fringe Counter.

In Figure 3.6a we assumed that

$$\Delta\Gamma_0 = 0, \quad \text{and } \Delta\tilde{\beta}C\Omega^{-2}Bk_c k_d > 0. \quad (3.5.3)$$

In Figure 3.6a it is clear that there is only one equilibrium point at the origin. It is also easy to see that Q-point drift, if large enough, could cause that equilibrium point to jump to a nonzero value, or have no equilibrium point at all.

The stability of the equilibrium points can be investigated using the approach in Section 3.2. In Figure 3.6b assume that the equilibrium point does not correspond to one of the jump discontinuities of the quantizer function. Then linearizing the system around this equilibrium point yields

$$\begin{bmatrix} \dot{x}_p \\ \dot{x}_c \end{bmatrix} = \begin{bmatrix} 0 & I & 0 \\ -\Omega^{-2} & -D & BC_c \\ 0 & 0 & A_c \end{bmatrix} \begin{bmatrix} \eta \\ \dot{\eta} \\ x_c \end{bmatrix}. \quad (3.5.4)$$

Thus, if the open loop system with the compensator is asymptotically stable, each of the equilibrium points will also be stable independent of the fringe frequency $\Delta\tilde{\beta}$.

To investigate the presence of limit cycles, the describing function analysis in the previous section can be used. The describing function for the quantization nonlinearity in (2.2.15) with Q-point drift is

$$N_{FC}(A) = 0, \quad 0 \leq A < \frac{L_f}{2}. \quad (3.5.5)$$

and

$$N_{FC}(A) = \frac{2L_f}{A^2\pi} \left\{ \sum_{m=1}^{M^+} \left[A^2 - \left(\frac{nL_f}{2} + \Delta\Gamma_0 \right)^2 \right]^{\frac{1}{2}} \right. \\ \left. \sum_{m=1}^{M^-} \left[A^2 - \left(\frac{-nL_f}{2} + \Delta\Gamma_0 \right)^2 \right]^{\frac{1}{2}} \right\} \quad (3.5.6)$$

where

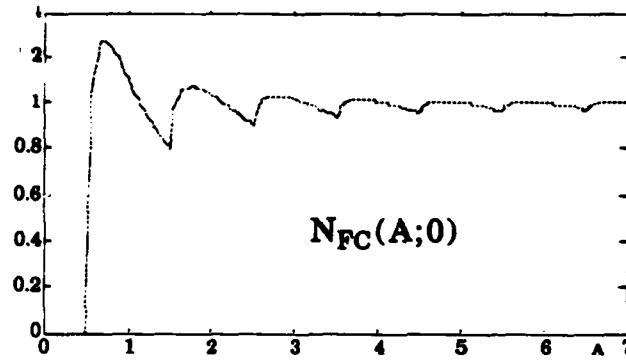
$$n = 2m - 1,$$

$$(M^+ - \frac{1}{2})L_f + \Delta\Gamma_0 < A < (M^+ + \frac{1}{2})L_f + \Delta\Gamma_0, \quad (3.5.7)$$

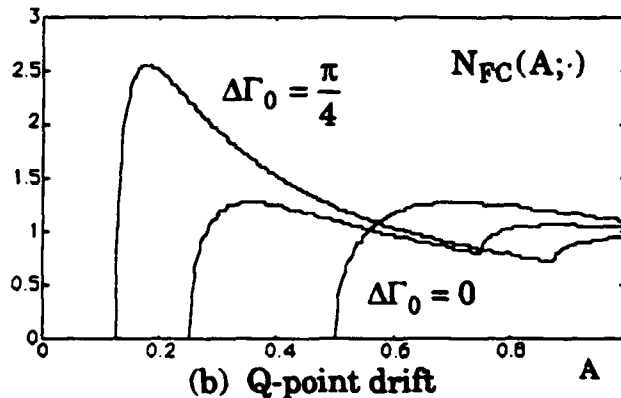
$$(M^- - \frac{1}{2})L_f - \Delta\Gamma_0 < A < (M^+ + \frac{1}{2})L_f - \Delta\Gamma_0.$$

A plot of this describing function for several values of $\Delta\Gamma_0$ is shown in Figure 3.8.

Proceeding as in the last section, we note that the describing function is always positive. Hence, if the root locus for the linear system is always in the left hand plane for $0 \leq k \leq k_d$, then this analysis does not predict any limit cycles for the fringe counting configuration of a modal domain optical fiber sensor. Limit cycles could occur, however, if the linear system is conditionally stable.



(a) $\Delta\Gamma_0 = 0$



(b) Q-point drift

Figure 3.7. Describing Function of the Quantizer Nonlinearity.

4. CONCLUSION

In this paper we have considered vibration suppression control systems for flexible structures which incorporate a modal domain optical fiber sensor. We have described the nonlinearities of these sensors as sinusoidal when the sensor is operated in its analog configuration and as a quantizer when the sensor is configured for fringe counting. The sinusoidal nonlinearity can introduce multiple equilibrium points and limit cycles. The fringe counting configuration can, under some circumstances, introduce limit cycles.

The analysis in this paper was carried out in terms of the parameters of the sensor, the fringe frequency and the Q-point drift. Given a structure with a modal domain optical fiber sensor, these results can be used to predict the nonlinear behavior in terms of potential disturbances, or to size the sensor to avoid undesirable nonlinear behavior.

ACKNOWLEDGMENT

This work was supported in part under ONR Contract N-00014-89-J-3123; in part under NASA Grant NAG-1-1006; and in part under NASA Grant NAG-1-1043.

REFERENCES

- Atherton, D.P., 1975. *Nonlinear Control Engineering*, Van Nostrand Reinhold Company:London.
- Blake, J. N., S. Y. Haung, B. Y. Kim, and H. S. Shaw, 1987. "Strain Effects on Highly Elliptical Core Two-Mode Fiber Devices," *Optics Letters*, Vol. 12, No. 9, pp. 732-734.
- Butter, C. D. and G. B. Hocker, 1978. "Fiber Optics Strain Sensor," *Appl. Opt.*, Vol. 18, pp. 1445-1447.
- Clark, R., C. Fuller, B. Fogg, W. Miller, A. Vengsarkar, and R. Claus, 1991. "Structural Control Using Optical Fiber Sensors and Piezoelectric Actuators," to appear in the Proceedings of the International Symposium & Exhibition on Active Materials & Adaptive Structures, Washington, DC..
- Cox, D. E. and D. K. Lindner, 1991. "Active Control for Vibration Suppression in a Flexible Beam Using a Modal Domain Optical Fiber Sensor," *ASME Journal of Vibration and Acoustics*, Vol. 113, pp. 369-382.
- Kim, B. Y., J. N. Blake, S. Y. Haung, and H. J. Shaw, 1987, "Use of Highly Elliptical Core Fibers for Two-Mode Fiber Devices," *Optics Letters*, Vol. 12, No. 9, pp. 729-731.

- Layton, M.R. and J. A. Bucaro, 1979. "Optical Fiber Acoustic Sensor Utilizing Mode-Mode Interference," *Applied Optics*, Vol. 18, pp. 666-670.
- Lindner, D. K., W. T. Baumann, K. M. Reichard, and F-S. Ho, 1991, "Spatial Filters for Control of Acoustic Radiation," submitted to the *ASME Journal of Vibration and Acoustics*.
- Lindner, D. K., K. M. Reichard, W. T. Baumann, and M. F. Barsky, 1990. "Measurement and Control of Flexible Structures Using Distributed Sensors." *Proceedings of the 29th IEEE Conf. on Decision and Control*, Honolulu, HI, pp. 2588-92.
- Lindner, D. K., W. T. Baumann, F. Ho, and E. Bielecki, 1991. "Modal Domain Optical Fiber Sensors for Control of Acoustic Radiation," *Proceedings of the Workshop on Recent Advances in Active Control of Sound and Vibration*, Blacksburg, VA, April, pp. 839-850.
- Murphy, K. A., M. S. Miller, A. M. Vengasarkar, and R. O. Claus, 1990. "Elliptical-Core Two Mode, Optical Fiber Sensor Implementation Methods," *Journal of Lightwave Technology*, Vol. 8, pp. 1688-1696.
- Reichard, K. M., 1991. "Distributed-Effect Modal Domain Optical Fiber Sensors for Flexible Structure Control," Ph.D. Thesis, Bradley Department of Electrical Engineering, Virginia Tech, Blacksburg, VA.
- Reichard, K. M. and Lindner, D. K., 1991. "Modeling the Effects of Arbitrary Stress on the Response of Modal Domain Optical Fiber Sensors," Internal Report, Bradley Department of Electrical Engineering, Virginia Tech, Blacksburg, VA.
- Sirkis, J. S. and H. W. Haslach, Jr., 1991. "Complete Phase-Strain Model for Structurally Embedded Interferometric Optical Fiber Sensors," *Journal of Intelligent Material Systems and Structures*, Vol 2, pp. 3-25.
- Snyder, A. and Love, J., 1983. *Optical Waveguide Theory*, Chapman and Hall, New York.

A Modified Gaussian Sum Approach to Estimation of NonGaussian Signals

Mauro J. Caputi
University of the Pacific
Dept. of Electrical & Computer Engr.
Stockton, CA 95211

Richard L. Moose
Virginia Polytechnic Inst. & S. U.
The Bradley Dept. of Electrical Engr.
Blacksburg, VA 24061-0111

Abstract

A Gaussian sum estimation algorithm has previously been developed to deal with noise processes that are nonGaussian. Inherent in this algorithm is a serious growing memory problem that causes the number of terms in the Gaussian sum to increase exponentially at each iteration. A modified Gaussian sum estimation algorithm is developed here that avoids the growing memory problem of the previous algorithm while providing effective state estimation. A simulation example is presented which illustrates the new nonGaussian estimation technique.

I. INTRODUCTION

Although the Gaussian assumption for modeling many types of signals and noise processes is valid in a wide range of applications, in practice it may not be a good assumption for some signals. Estimation techniques therefore need to be developed that can cope effectively with the nonGaussian nature of certain signals. One such approach previously developed by Sorenson and Alspach [1] is the Gaussian sum technique. The density function of each nonGaussian process of the system is approximated by a

This work was supported by the Office of Naval Research, grant N00014-89-J-3123.

weighted sum of Gaussian density functions. The conditional density of the state given the available measurement sequence, necessary in the Kalman filter development, is updated using the Gaussian sum approximations and Bayes' rule.

A serious limitation in the approach used by Sorenson and Alspach is that the number of Gaussian terms used to approximate the density functions increases at each time iteration. An alternate approach is required to alleviate this limitation. A modified estimation algorithm is developed here based on an adaptive Kalman filter scheme first presented by Magill [2], and extended by Moose [3]. Essentially, a parameter vector is used to uniquely describe each Gaussian term in the estimator. The parameter vector is restricted to be randomly chosen from the same finite set of known values at each iteration. By using a nonGaussian signal model in conjunction with the modified formulation of the Gaussian sum estimator, the number of Gaussian terms at each iteration of the estimator will be fixed, thereby avoiding the growing memory problem.

The nonGaussian signal model and associated Gaussian sum density approximation is developed in section II. The modified estimation algorithm, termed the modified Gaussian sum (MGS) adaptive filter [4], is developed in section III. A simulation example implementing the MGS adaptive filter is presented in section IV. The conclusions are given in section V.

II. NONGAUSSIAN SIGNAL MODEL DEVELOPMENT

Let u be a random noise process or random input signal with a nonGaussian density function. It can be modeled as the sum of two statistically independent random processes

$$u = b + n \quad (1)$$

The first term, b , is a semi-Markov process with state transitions governed by the transition probability matrix of a conventional Markov process. Markov processes have the property that a transition is made at every time instant. The transition may return the process to the state it previously occupied, but a transition occurs nevertheless. However, in the semi-Markov case, the amount of time between transitions is a random variable [5]. The value of b is randomly selected from a fixed set of discrete values, characterized by a delta probability density function

$$p(b) = \sum_{i=1}^M P_i \delta(b - b_i) \quad (2)$$

with

$$\sum_{i=1}^M P_i = 1; \quad P_i \geq 0 \text{ for } i = 1, 2, \dots, M \quad (3)$$

This process can be thought of as a randomly-switching bias, each bias value b_i having probability P_i .

The second term, n , is a zero mean white Gaussian process with variance σ_n^2 . With both densities known, the density function of u can be found using the convolution relationship between u , n , and b [6]

$$p(u) = \int_{-\infty}^{\infty} p_n(u - b) p_b(b) db \quad (4)$$

where $p_n(u - b)$ is the Gaussian density with $n = u - b$

$$p_n(u - b) = \frac{1}{\sqrt{2\pi}\sigma_n} e^{-\frac{1}{2}\left(\frac{u-b}{\sigma_n}\right)^2} \quad (5)$$

Substituting (2) and (5) into the convolution integral (4) gives

$$p(u) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_n} e^{-\frac{1}{2}\left(\frac{u-b}{\sigma_n}\right)^2} \sum_{i=1}^M P_i \delta(b-b_i) db \quad (6)$$

Interchanging integration and summation

$$p(u) = \sum_{i=1}^M P_i \left[\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_n} e^{-\frac{1}{2}\left(\frac{u-b}{\sigma_n}\right)^2} \delta(b-b_i) db \right] \quad (7)$$

Using the sifting property of the delta function [7]

$$f(a) = \int_{-\infty}^{\infty} f(x) \delta(x-a) dx \quad (8)$$

the integral of (7) is evaluated as

$$p(u) = \sum_{i=1}^M P_i \left[\frac{1}{\sqrt{2\pi}\sigma_n} e^{-\frac{1}{2}\left(\frac{u-b_i}{\sigma_n}\right)^2} \right] \quad (9)$$

or

$$p(u) = \sum_{i=1}^M P_i N[b_i, \sigma_n^2] \quad (10)$$

Thus, the nonGaussian density function of u can be modeled as a Gaussian sum. The weight P_i of each Gaussian term is the probability of the i^{th} bias term. The bias term b_i is restricted to be randomly selected from the same fixed set of bias values at each iteration. Using this model in conjunction with the modified Gaussian sum adaptive filter developed in the next section avoids the growing memory problem of Sorenson and Alspach's development.

Selecting the parameters P_i , b_i , and σ_n in (10) to obtain the "best" approximation p_{GS} to some actual nonGaussian density function p_A is accomplished by means of minimizing the L^k norm

$$\|p_A - p_{GS}\|^k = \int_{-\infty}^{\infty} \left| p_A(u) - \sum_{i=1}^M P_i N[b_i, \sigma_n^2] \right|^k du \quad (11)$$

This curve fitting exercise can be done off-line using several values of M until a suitable trade-off between minimum norm and minimum M is obtained. Sorenson and Alspach [1] performed this curve fitting procedure using L^1 and L^2 norms for a uniform density and a Gamma density. It was found that minimizing the L^2 norm resulted in many fewer terms in the Gaussian sum and a considerably better looking approximation for both densities compared to minimizing the L^1 norm.

Fig. 1 compares a Gamma density with a four-term Gaussian sum density approximation minimizing the L^2 norm. The Gaussian sum curve is shown to fit the Gamma curve reasonably well. The Gamma density used is

$$p(u) = \begin{cases} \frac{u^3 e^{-u}}{6} & u \geq 0 \\ 0 & u < 0 \end{cases} \quad (12)$$

Each term of the Gaussian sum has a fixed value of $\sigma_n = 1$. Table I lists the values of P_i and b_i used in the Gaussian sum.

TABLE I.
Gaussian sum P_i , b_i values, $\sigma_n = 1$.

| i | P_i | b_i |
|-----|-------|-------|
| 1 | 0.081 | 2.537 |
| 2 | 0.432 | 2.553 |
| 3 | 0.356 | 4.555 |
| 4 | 0.131 | 6.933 |

III. MODIFIED GAUSSIAN SUM ADAPTIVE FILTER

A modified Gaussian sum adaptive filter is now developed for a linear system with a deterministic input signal, nonGaussian plant noise, and nonGaussian measurement noise. The system is modeled in standard discrete-time state-space form as

$$x_{k+1} = \Phi x_k + \Gamma u_k + \Psi w_k \quad (13)$$

$$w_k = a_k + m_k \quad (14)$$

$$z_k = H x_k + v_k \quad (15)$$

$$v_k = b_k + n_k \quad (16)$$

where x_{k+1} is the state vector
 u_k is a known deterministic input
 w_k is the vector Gaussian sum signal model of the actual nonGaussian plant noise process, comprised of semi-Markov bias vector a_k , and zero mean white Gaussian noise m_k with covariance Q
 z_k is the measurement vector
 v_k is the vector Gaussian sum signal model of the actual nonGaussian measurement noise process, comprised of semi-Markov bias vector b_k , and zero mean white Gaussian noise n_k with covariance R

Φ, Γ, Ψ, H are the respective constant transition matrices

and the random quantities x_k, a_k, m_k, b_k, n_k are assumed to be mutually statistically independent.

The optimal estimate \hat{x}_{k+1} of the state vector is found by minimizing the mean-

square error between x_{k+1} and \hat{x}_{k+1} . This results in the conditional mean estimate

$$\hat{x}_{k+1} = E[x_{k+1} | Z_{k+1}] = \int_{-\infty}^{\infty} x_{k+1} p(x_{k+1} | Z_{k+1}) dx_{k+1} \quad (17)$$

where Z_{k+1} is the current measurement sequence $\{z_1, z_2, \dots, z_{k+1}\}$. The conditional density function of (17) can be written as the ratio of the corresponding joint and marginal densities

$$p(x_{k+1} | Z_{k+1}) = \frac{p(x_{k+1}, Z_{k+1})}{p(Z_{k+1})} \quad (18)$$

The two bias vectors a and b are explicitly brought into (18) by considering the joint density $p(x_{k+1}, Z_{k+1})$ to be a marginal density found from $p(x_{k+1}, Z_{k+1}, a_{k+1} = a_i, b_{k+1} = b_j)$ by summing over the a and b terms

$$p(x_{k+1} | Z_{k+1}) = \frac{\sum_{i=1}^N \sum_{j=1}^M p(x_{k+1}, Z_{k+1}, a_{k+1} = a_i, b_{k+1} = b_j)}{p(Z_{k+1})} \quad (19)$$

Using Bayes' rule and using $p(x_{k+1}, Z_{k+1}, a_i, b_j)$ as shorthand for $p(x_{k+1}, Z_{k+1}, a_{k+1} = a_i, b_{k+1} = b_j)$, the conditional density of (19) becomes

$$p(x_{k+1} | Z_{k+1}) = \sum_{i=1}^N \sum_{j=1}^M p(x_{k+1} | Z_{k+1}, a_i, b_j) p(a_i, b_j | Z_{k+1}) \quad (20)$$

Substituting (20) into the conditional mean (17), and interchanging integration and summation results in

$$\hat{x}_{k+1} = \sum_{i=1}^N \sum_{j=1}^M \left[\int_{-\infty}^{\infty} x_{k+1} p(x_{k+1} | Z_{k+1}, a_i, b_j) dx_{k+1} \right] p(a_i, b_j | Z_{k+1}) \quad (21)$$

The bracketed integral in (21) is the conditional mean estimate of x_{k+1} given that $a_{k+1} = a_i$ and $b_{k+1} = b_j$, denoted by

$$\hat{x}_{k+1}^{ij} = \int_{-\infty}^{\infty} x_{k+1} p(x_{k+1} | Z_{k+1}, a_i, b_j) dx_{k+1} \quad (22)$$

In effect, \hat{x}_{k+1}^{ij} represents the estimate for the ij^{th} density combination from the two Gaussian sums. A fixed set of $N \times M$ (NM) estimators is needed to provide all of the individual \hat{x}_{k+1}^{ij} estimates. The overall estimate from (21) and (22),

$$\hat{x}_{k+1} = \sum_{i=1}^N \sum_{j=1}^M \hat{x}_{k+1}^{ij} p(a_i, b_j | Z_{k+1}) \quad (23)$$

is a weighted sum of the NM individual estimates. The weighting factor $p(a_i, b_j | Z_{k+1})$ is the probability that $a_{k+1} = a_i$ and $b_{k+1} = b_j$ given the current measurement sequence Z_{k+1} . Since the number of terms in the overall estimate (23) is fixed, the growing memory problem of [1] is avoided.

Each estimate (22) is found from a Kalman filter equation, modified [4] to include the two bias terms a_i and b_j of the Gaussian sum densities, given by

$$\hat{x}_{k+1}^{ij} = \Phi \hat{x}_k^{ij} + \Gamma u_k + \Psi a_i + K_{k+1} [z_{k+1} - b_j - H(\Phi \hat{x}_k^{ij} + \Gamma u_k + \Psi a_i)] \quad (24)$$

The Kalman gain and covariance terms

$$M_{k+1} = \Phi P_k \Phi^T + \Psi Q \Psi^T \quad (25)$$

$$K_{k+1} = M_{k+1} H^T (H M_{k+1} H^T + R)^{-1} \quad (26)$$

$$P_{k+1} = (I - K_{k+1} H) M_{k+1} \quad (27)$$

are the same for each i_j^{th} estimate (24) because the covariances Q and R remain fixed for each respective bias parameter a_i and b_j .

The weighting term $p(a_i, b_j | Z_{k+1})$ of (24) is found next. Using $Z_{k+1} = \{Z_k, z_{k+1}\}$ and Bayes' rule, the weighting term becomes

$$p(a_i, b_j | Z_{k+1}) = \frac{p(z_{k+1} | a_i, b_j, Z_k) p(a_i, b_j | Z_k)}{p(z_{k+1} | Z_k)} \quad (28)$$

The first term of the numerator of (28) can be approximated by a Gaussian density if the bias terms switch slowly compared to the time interval k [3], given by

$$p(z_{k+1} | a_i, b_j, Z_k) = N[H(\Phi \hat{x}_k^{ij} + \Gamma u_k + \Psi a_i) + b_j, H M_{k+1} H^T + R] \quad (29)$$

with mean and covariance terms available from the Kalman filter equation (24) and Kalman gain (26).

The second term of the numerator of (28) is the predicted probability value that $a_{k+1} = a_i$ and $b_{k+1} = b_j$ given the past measurement sequence Z_k , given by

$$p(a_{k+1} = a_i, b_{k+1} = b_j | Z_k) = \sum_{\alpha=1}^N \sum_{\beta=1}^M \theta_a^{i\alpha} \theta_b^{j\beta} w_k^{\alpha\beta} \quad (30)$$

where

$$\theta_a^{i\alpha} = p(a_{k+1} = a_i | a_k = a_\alpha) \quad (31)$$

$$\theta_b^{j\beta} = p(b_{k+1} = b_j | b_k = b_\beta) \quad (32)$$

$$w_k^{\alpha\beta} = p(a_k = a_\alpha, b_k = b_\beta | Z_k) \quad (33)$$

The $\theta_a^{i\alpha}$ and $\theta_b^{j\beta}$ terms are Markov transition probabilities [5]; that is, $\theta_a^{i\alpha}$ is the conditional probability that $a = a_i$ at time interval $k+1$, given that $a = a_\alpha$ at time

interval k . The $\theta_b^{j\beta}$ term is similarly defined. The term $w_k^{\alpha\beta}$ is of the same form as (28) and is just the previous weighting term at the previous time interval k .

The denominator term of (28) is independent of ij . Therefore it is the same for each ij^{th} estimator and becomes a scale factor.

Combining (29) - (33), the weighting term (28) is written as [4]

$$w_{k+1}^{ij} = p(a_i, b_j | Z_{k+1}) = C_{k+1}^o p(z_{k+1} | a_i, b_j, Z_k) \sum_{\alpha=1}^N \sum_{\beta=1}^M \theta_a^{i\alpha} \theta_b^{j\beta} w_k^{\alpha\beta} \quad (34)$$

where C_{k+1}^o is a scale factor determined at each time interval such that

$$\sum_{i=1}^N \sum_{j=1}^M w_{k+1}^{ij} = 1 \quad (35)$$

guaranteeing that the sum of all the weighting terms (34) is equal to one.

The structure of the overall MGS adaptive filter is a fixed bank of NM Kalman filters operating in parallel, with each individual estimate multiplied by its own corresponding weighting term. The ij^{th} estimator based on the bias terms that most closely matches the actual bias terms of the modeled system will have a corresponding weighting term that tends closer to one, while the weights of the other mismatched estimators will tend towards zero. A block diagram of the MGS adaptive filter is shown in Fig. 2.

IV. SIMULATION EXAMPLE

An example illustrating the modified Gaussian sum estimation technique is now presented. A first-order system is used, modeled by the following discrete-time equations

$$x_{k+1} = e^{-\alpha T} x_k + (1 - e^{-\alpha T}) u_k + \frac{(1 - e^{-\alpha T})}{\alpha} w_k \quad (36)$$

$$w_k = a_k + m_k \quad (37)$$

$$z_k = x_k + v_k \quad (38)$$

$$v_k = b_k + n_k \quad (39)$$

The value of α is 0.6 and the sample time $T = 1$ second. For simplicity, let the plant noise (37) be zero mean white Gaussian with variance $Q = 1.0$. Therefore, the randomly-switching plant bias term takes on the value of $a_k = 0$. The MGS adaptive filter structure now reduces to a bank of only M filters operating in parallel. The actual measurement noise, modeled by (39), has the Gamma density of (12), with a mean and variance of 4. The measurement bias term b_k can be randomly selected from the last three bias terms of TABLE I, {2.553, 4.555, 6.933}. The first bias term of TABLE I, {2.537}, is not used since it is so close in value to the second bias term. The measurement model noise term n_k is zero mean white Gaussian with variance $R = 1.0$. A deterministic input of $u_k = 10$ is used throughout the simulation. Fig. 3a shows the measurement and state sequences. Note how the measurement is centered about 14, indicating a mean value for the Gamma density of 4.

The filter is initialized with equally-valued weighting terms

$$w_0^j = \frac{1}{3}, \quad \text{for } j = 1, 2, 3 \quad (40)$$

A Markov transition probability matrix $\Theta_b^{j\beta}$, consisting of $\theta_b^{j\beta}$ elements, is configured with a high probability that the bias term does not switch from one value to another, and a low probability that the bias term does switch, given by

$$\Theta_b^{j\beta} = \begin{bmatrix} .95 & .025 & .025 \\ .025 & .95 & .025 \\ .025 & .025 & .95 \end{bmatrix} \quad (41)$$

The initial value of the state and state estimate is $x_0 = \hat{x}_0 = 20$, and the initial value of the variance of the error $x_{k+1} - \hat{x}_{k+1}$ is $P_o = 100$. The overall state estimate and the state are shown in Fig. 3b, with the error and overall state estimate shown in Fig. 3c. Note how the error appears to be zero mean, thus showing that the MGS adaptive filter removes the bias effect of the nonzero mean Gamma measurement noise.

Figs. 4a, b, and c show the weighting terms for each of the b_k bias terms. In order to lessen the noise of the weighting terms, a first-order lowpass filter

$$w_{k+1}^j = \lambda w_k^j + (1 - \lambda)w_{k+1}^j \quad (42)$$

is used to smooth the weighting terms, where $\lambda = 0.7$. Fig. 4d can be thought of as the overall measurement bias estimate \hat{b}_k due to the nonzero mean Gamma measurement noise. Using (24) and (34), this overall bias estimate is part of the overall state estimate (23) and is written in this case with $a_k = 0$ as

$$\hat{b}_k = \sum_{j=1}^M w_k^j b_k^j \quad (43)$$

Note how this overall bias estimate approximately models the mean value of 4 of the Gamma measurement noise.

V. CONCLUSIONS

An estimation technique has been developed which can cope effectively with nonGaussian signals. This MGS adaptive filter is comprised of a fixed set of estimators operating in parallel with each individual estimate possessing its own corresponding weighting term. The MGS adaptive filter is suitable for practical implementation since it avoids the growing memory problem of a previously developed algorithm.

REFERENCES

- [1] Sorenson, H. W., and Alspach, D. L. (1971)
Recursive Bayesian estimation using Gaussian sums.
Automatica, vol. 7, 1971, 465-479.
- [2] Magill, D. T. (1965)
Optimal adaptive estimation of sampled stochastic processes.
IEEE Transactions on Automatic Control, AC-10 (Oct. 1965), 434-439.
- [3] Moose, R. L. (1975)
An adaptive state estimation solution to the maneuvering target problem.
IEEE Transactions on Automatic Control, AC-20 (June 1975), 359-362.
- [4] Caputi, M. J. (1991)
NonGaussian estimation using a modified Gaussian sum adaptive filter.
Ph.D. dissertation, Bradley Dept. of Electrical Engineering,
Virginia Polytechnic Institute & State University, Blacksburg, 1991.
- [5] Howard, R. A. (1971)
Dynamic Probabilistic Systems, Volume II: Semi-Markov and Decision Processes.
New York: John Wiley and Sons, Inc., 1971.
- [6] Papoulis, A. (1991)
Probability, Random Variables, and Stochastic Processes, 3rd ed.
New York: McGraw-Hill, Inc., 1991.
- [7] Stremler, F. G. (1990)
Introduction to Communication Systems, 3rd ed.
Addison-Wesley, 1990.

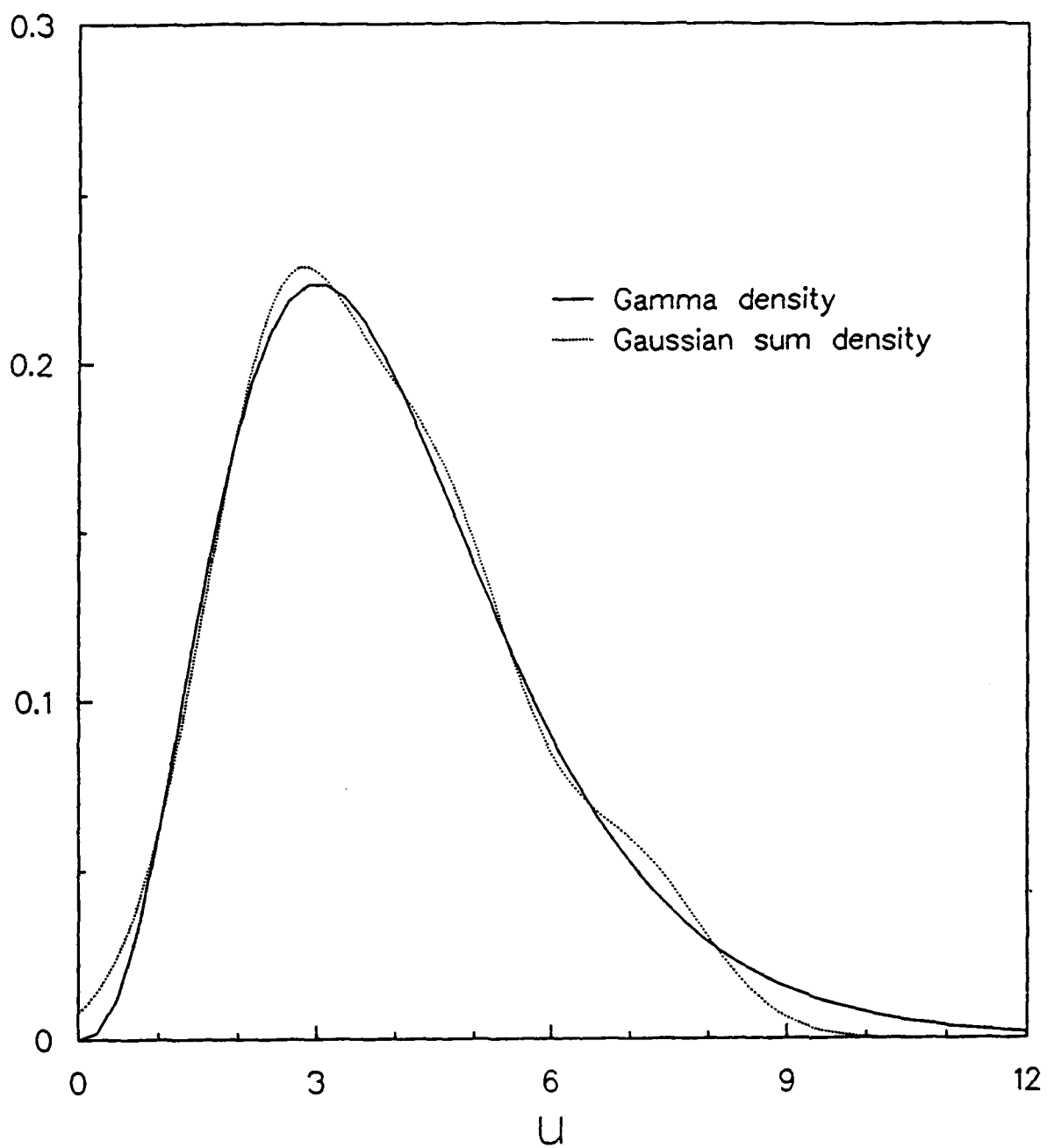


Fig. 1. Gamma and Gaussian sum probability density functions.

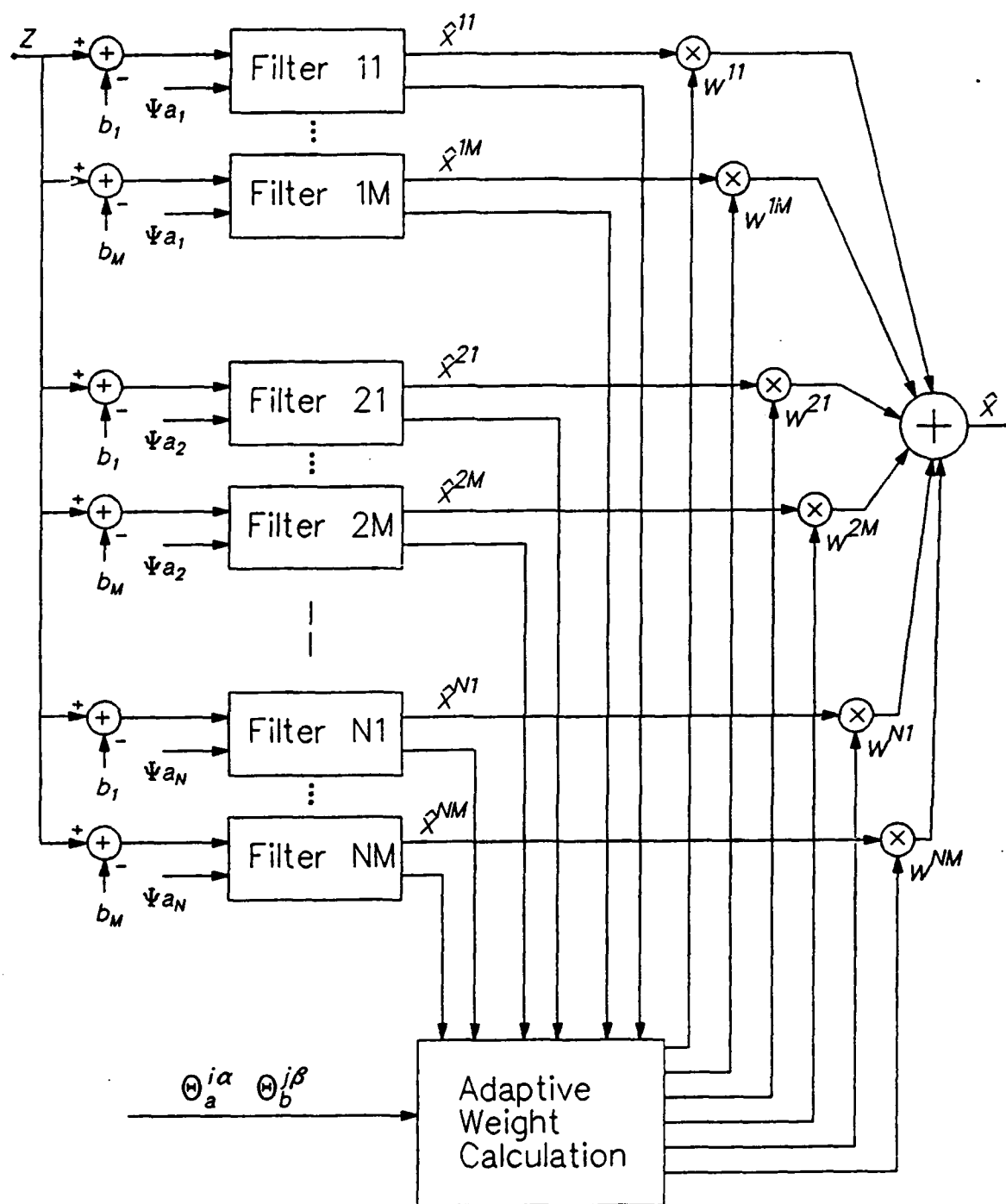


Fig. 2. Modified Gaussian sum adaptive filter structure.

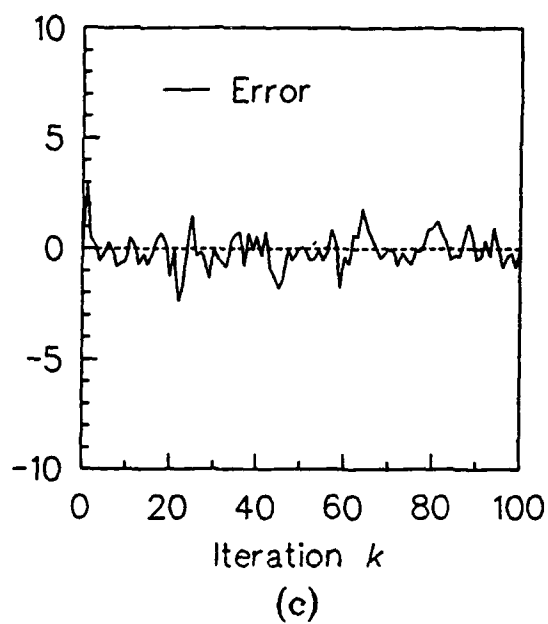
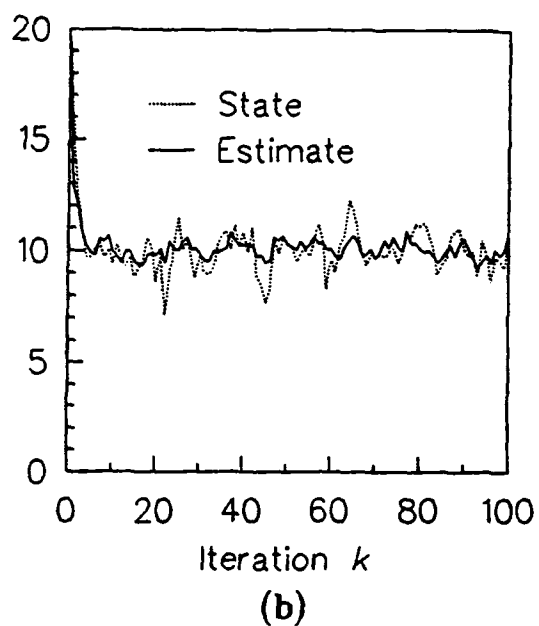
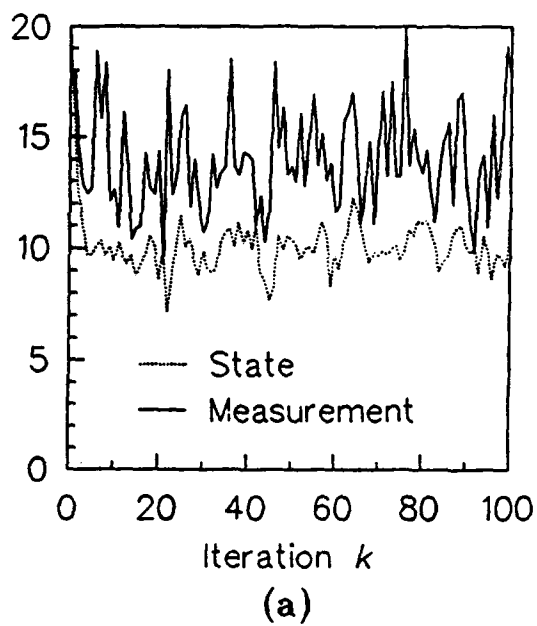


Fig. 3. (a) Measurement and state simulation, (b) modified Gaussian sum estimate compared with state, (c) error = state - estimate.

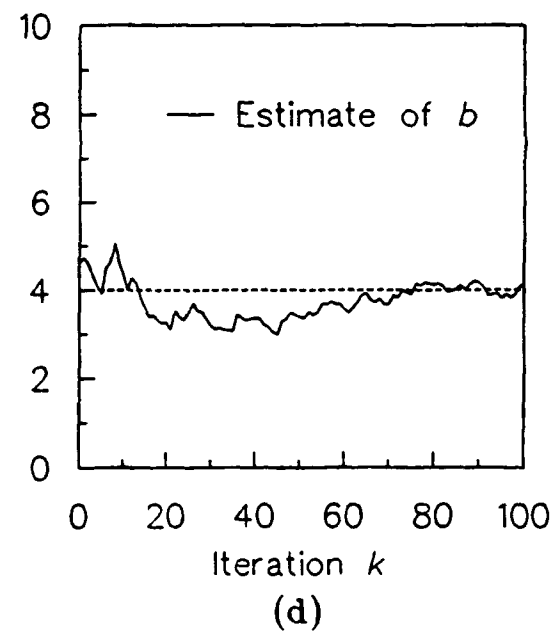
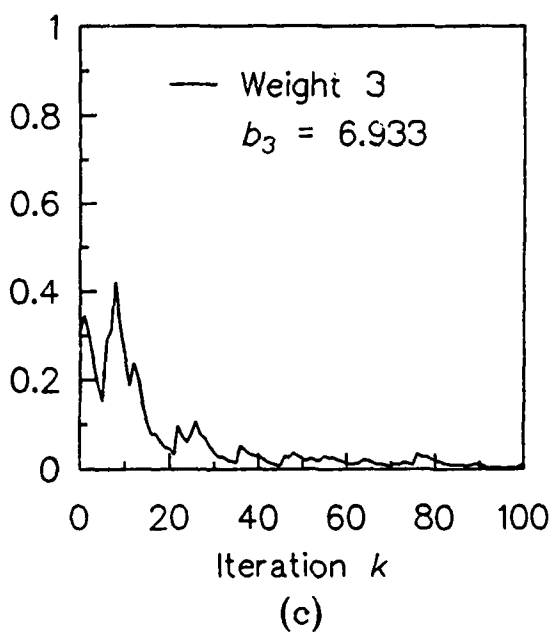
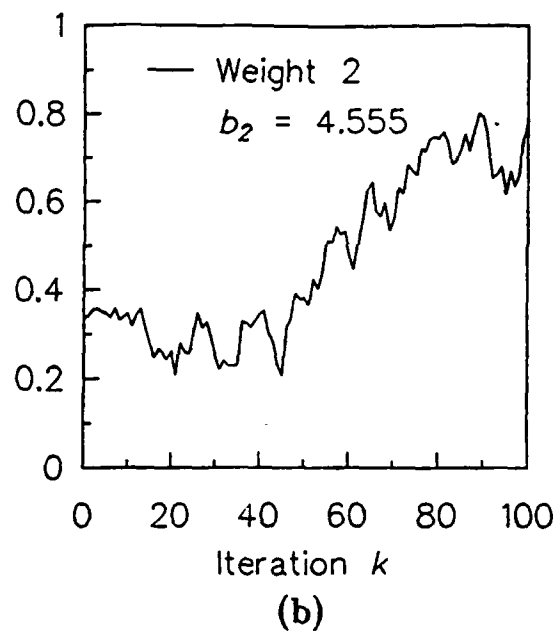
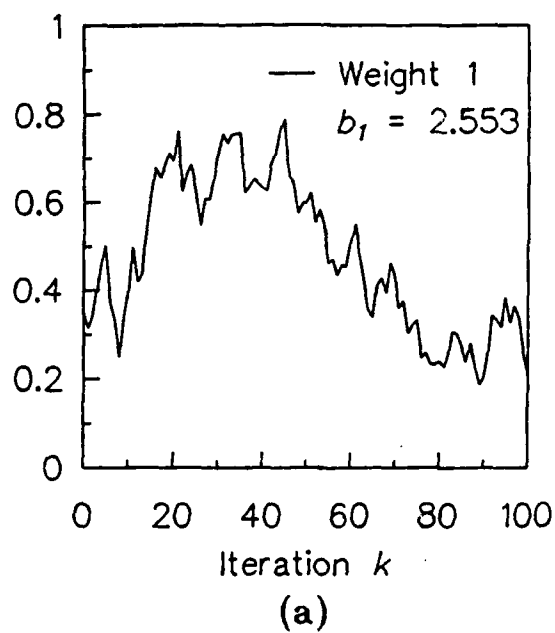


Fig. 4. (a) Weight 1, (b) weight 2, (c) weight 3, (d) overall measurement bias estimate.

MEASUREMENT AND CONTROL OF FLEXIBLE STRUCTURES USING DISTRIBUTED SENSORS

Douglas K. Lindner, Karl M. Reichard, William T. Baumann

Fiber & Electro-Optics Research Center
Bradley Department of Electrical Engineering
Virginia Polytechnic Institute & State University
Blacksburg, VA 24061

Michael F. Barsky

Hughes Aircraft Company
Los Angeles, CA

Abstract

This paper describes the use of distributed filtering to realize arbitrary scalar system outputs from the output of a distributed sensor. The set of scalar position measurements realizable using traditional point sensors is a subset of the set of scalar outputs realizable using distributed sensing and filtering. This is one of the advantages of distributed sensing over point sensing. Several examples are provided to illustrate applications of distributed filtering. It is shown that distributed-effect sensors can be used to implement functional observers, providing a significant reduction in compensator order.

1. Introduction

Sensor placement continues to be an important issue in the control of flexible structures. Restrictions on sensor placement often result in limitations on the set of realizable system outputs or measurements - particularly when the outputs or measurements are produced by point sensors located on the structure.

Distributed sensors provide an attractive solution to many of the problems associated with the use of point sensors in flexible structure control. Distributed sensors produce a spatially continuous measurement of the system behavior. Most control systems, however, employ scalar signal processing and computing technology; therefore, distributed sensor outputs are usually converted into scalar signals before they are made available to the control system. The challenge is to retain in the scalar signal, as much of the information contained in the distributed signal as possible. The process of converting the distributed measurement into a scalar signal is essentially distributed filtering.

The work presented in this paper provides the theoretical foundation for the use of distributed sensing and filtering in a variety of control applications. This technology is already being applied experimentally, particularly in the area of modal filtering [4,3]. The idea of filtering distributed signals to provide scalar sensor outputs also applies to the design of distributed-effect sensors - sensors that produce a scalar output that is based on a distributed measurement [2].

In this paper, we consider the application of distributed sensing and filtering to the class of one-dimensional flexible structures represented by flexible beams, and it is assumed that there are no restrictions on the filter functions that may be implemented. In section 2 we describe mathematical models for the class of systems and sensors considered in this paper. In the third section, we describe the filtering process and show the existence of filter functions for realizing any arbitrary scalar system output consisting of a linear combination of the system position states. In section 4, we present several examples of distributed filtering applications, including the design of low-order compensators. The final section contains a summary of the results presented in this paper and describes some areas for future research.

2. System and sensor models

In this section we describe the mathematical model for the sensor and a mathematical model for the class of flexible structures considered in this paper. Since we are concerned with the application of distributed sensing to the class of flexible structures subject to bending and vibrations in one dimension, we consider as a representative example the model for a flexible beam.

2.1 System model

Let $y(z,t)$ denote the deflection of the flexible structure from equilibrium (defined as $y(z,t)=0$) at the point z in the domain of the structure, $D(\mathcal{V})=\{z | 0 \leq z \leq l\}$, at time t . For $z \in D(\mathcal{V})$, $y(z,t)$ describes the shape of the structure as a function of time. We shall assume there exists a set of functions $\{\phi_j(z)\}$ which form an orthonormal basis for $L^2(\mathcal{V})$, the set of all square-integrable functions defined on $D(\mathcal{V})$ [5]. At any time $t=t_0$, the set of all structural shapes $y(z,t)$ is contained in $L^2(\mathcal{V})$; therefore, given any shape $y(z,t)$, there exists a set of time varying weights $\{\eta_j(t)\}$ such that

$$y(z,t) = \sum_{j=1}^{\infty} \eta_j(t) \phi_j(z). \quad (1)$$

The basis functions $\{\phi_j(z)\}$ are commonly referred to as mode shapes of the structure, and the time varying $\{\eta_j(t)\}$ are called the modal amplitudes or modal weight.

While $y(z,t)$ in (1) is expressed as an infinite sum of mode shapes, we typically assume that $y(z,t)$ can be reasonably approximated by a finite number of modes, so that

$$y(z,t) = \sum_{j=1}^N \eta_j(t) \phi_j(z). \quad (2)$$

For models of flexible structures, the upper limit N in (2) may be very large.

Given the basis $\{\phi_j(z)\}$, the modal amplitudes $\{\eta_j(t)\}$ completely describe the shape of the structure, $y(z,t)$. In the modal basis, one mathematical model of the flexible structure is described by a second-order ordinary differential equation of the form

$$M\ddot{\eta}(t) + K\dot{\eta}(t) = B u(t), \quad (3)$$

where

$$\eta(t) = [\eta_1(t) \ \eta_2(t) \ \cdots \ \eta_N(t)]^T. \quad (4)$$

The operators M , K , and B in (3) are known as the mass, stiffness, and input influence matrices respectively, and $u(t)$ denotes external forces (control or disturbance) which may act on the structure.

Let $z(t)$ denote a system output or measurement. System position outputs are modelled as linear combinations of the system position states in (3) - that is, there exists an output or measurement matrix C of appropriate dimensions such that

$$z(t) = C\eta(t). \quad (5)$$

In the special case where $z(t)$ is a position measurement that corresponds to the output of a point sensor located at a point $x = x_0$ on the structure, $z(t) = y(x_0, t)$ and the output matrix C in (5) is given by

$$C = [\phi_1(x_0) \phi_2(x_0) \dots \phi_N(x_0)], \quad (6)$$

where $\phi_j(x_0)$ denotes the value of the j -th mode shape evaluated at the point $x = x_0$.

2.2 Sensor and filter models

The output of an ideal distributed position sensor is simply the structural shape $y(x, t)$. An example of such a sensor is the holographic sensor described in [1,2]. For structural displacements within the linear range of the sensor, the spatial variation in the intensity of the optical signal is proportional to the shape of the structure at time t [1].

To convert the information provided by the distributed sensor into a form that can be processed by the control system, one or more scalar signals, $e(t)$, are formed by performing the spatial filtering operation

$$e(t) = \int_0^l K(x) y(x, t) dx. \quad (10)$$

We call $K(x)$ the filter transmittance or weighting function, and refer to $e(t)$ as the filtered output of the distributed sensor. In the case of the holographic sensor this is accomplished by passing the output through an optical filter with a spatially varying optical transmittance, $K(x)$, before it is processed by a photodetector. The photodetector produces an output proportional to the spatial integral of the intensity of the incident optical signal.

In a distributed-effect sensor, the output of the sensor is a spatially filtered function of $y(x, t)$, but the function $y(x, t)$ is not directly available. An example of this type of sensor is a modal-domain optical fiber sensor. An optical fiber attached to a beam is sensitive to strain along its length. If the sensitivity of the fiber to strain can be varied as a function of position along the fiber, then, ignoring some constants, the output of a photodetector at the far end of the fiber can be assumed to be

$$e(t) = \int_0^l K(x) \frac{\partial^2 y}{\partial x^2}(x, t) dx. \quad (11)$$

The net effect of spatially filtering and integrating the output of a distributed sensor, but a distributed output is never directly available.

3. Filter design

For simplicity, we will restrict our attention, in this section, to distributed position sensors, but the results may be extended to other distributed sensors, such as those that measure strain or velocity. If the system output $z(t)$ in (5) corresponds to the output $e(t)$ in (10), then

$$\int_0^l K(x) y(x, t) dx = C\eta(t). \quad (12)$$

We need to answer the following question: Given an arbitrary output matrix C , does there exist a filter weighting function $K(x)$, such that (12) holds? If the answer to this question is yes, then using distributed sensor output filtering, we can realize any desired scalar system position output or measurement.

Substituting (2) into (10) and interchanging the order of integration and summation, the output may be written as

$$e(t) = \sum_{j=1}^N \left(\int_0^l K(x) \phi_j(x) dx \right) \eta_j(t). \quad (13)$$

Because the mode shapes $\{\phi_j(x)\}$ are a basis for the set of all square-integrable functions defined on $D(\mathcal{V})$ [5], they are also a basis for the set of all square-integrable filter transmittance functions. If we restrict $K(x)$ to the finite-dimensional space of square-integrable functions spanned by the set of N mode shapes used to approximate $y(x, t)$ in (2), then there exists a set of N scalars $\{k_1, \dots, k_N\}$ such that

$$K(x) = \sum_{i=1}^N k_i \phi_i(x). \quad (14)$$

Substituting the modal representation (14) for $K(x)$ into the expression (13) for the output yields

$$e(t) = \sum_{j=1}^N \left(\int_0^l \sum_{i=1}^N k_i \phi_i(x) \phi_j(x) dx \right) \eta_j(t). \quad (15)$$

Interchanging the order of summation and integration in (15), and recalling that the mode shapes are orthonormal yields

$$e(t) = \sum_{j=1}^N k_j \eta_j(t). \quad (16)$$

Equation (16) implies that for $z(t) = e(t)$, as in (12), the output matrix C is given by

$$C = [k_1 \ k_2 \ \dots \ k_N]. \quad (17)$$

Conversely, for an arbitrary output matrix C with elements c_j , if

$$K(x) = \sum_{j=1}^N c_j \phi_j(x), \quad (18)$$

then the filtered output of the distributed sensor, $e(t)$, can be written in the form (5), and equation (12) is satisfied.

4. Examples

4.1 Point sensing

One application of distributed sensors is to duplicate the output of a point sensor located somewhere on the structure. The output matrix C for $z(t) = y(x_0, t)$, was given in (6). By (18), the corresponding weighting function is

$$K(x) = \sum_{j=1}^N \phi_j(x_0) \phi_j(x). \quad (19)$$

Setting the scalar output $z(t)$ equal to the output $e(t)$ in (11), yields

$$y(x_0, t) = \int_0^l K(x) y(x, t) dx, \quad (20)$$

which implies

$$K(x) = \delta(x - x_0), \quad (21)$$

where $\delta(x - x_0)$ is the Dirac delta function centered at $x = x_0$.

This example shows that the set of filter weighting functions that produce scalar outputs corresponding to the outputs of point sensors is equal to the set of delta functions centered at points $x_0 \in D(\mathcal{V})$. Clearly, this set is only a subset of the set of all admissible filter weighting functions; therefore, the set of position output matrices realizable using point sensors, is a subset of the set of output matrices realizable using distributed sensor filtering.

4.2 Modal filtering

Another application of distributed sensors is modal filtering. The filter weighting function $K(x)$ may be chosen to eliminate certain modal components, or modes, from the system output. This is analogous to the familiar notion of band-limited filtering in conventional signal processing. The concept of distributed modal filtering for flexible structures has been demonstrated using distributed-effect sensors made of piezoelectric film [4], and was used to explain the robustness properties of an LQG control system for a flexible beam using the holographic sensor [3].

The usefulness of distributed sensors for modal filtering is a result of the orthogonality of the mode shapes that are the basis for the set of filter weighting functions and the structure shape functions. Choosing a set of modes, such as the set $B = \{\phi_{j_m}(x) | 0 \leq m \leq M\}$, and selecting a filter weighting function

$$K(x) = \sum_{m=0}^M k_m \phi_{j_m}(x), \quad (22)$$

composed only of modes contained in B , the filtered distributed sensor output is

$$z(t) = \sum_{j_m=0}^M k_m \eta_{j_m}(t). \quad (23)$$

Because the mode shapes are mutually orthogonal, any modes contained in the distributed measurement $y(x,t)$ but not contained in the set B are removed by the filter and do not appear in the filtered output. Modal filtering is particularly useful for reducing observation spillover when the control system is designed using a reduced order model of the system since it permits removal of any unmodelled system dynamics from the measurement signal.

4.3 Distributed Compensator Gains

To control a system described by the second-order differential equation (3) using a single actuator, a standard approach is to rewrite the equation in first order form

$$\begin{aligned} \dot{z} &= A z + B u \\ z &= C z \end{aligned} \quad (24)$$

using the state $z = [\eta^T \dot{\eta}^T]^T$. A state feedback gain is computed so that the control $u = -kz$ provides the desired performance. Then, since the entire state is not available for feedback, a state estimator is constructed to provide an estimate of z , denoted \hat{z} , and the feedback $u = -k\hat{z}$ is used. A full-order state estimator would be of dimension $2N$ and have the form

$$\dot{\hat{z}} = (A - LC)\hat{z} + B u + L z. \quad (25)$$

If z is of dimension m , then a reduced-order observer of dimension $2N-m$ can be used to provide an estimate of z . To reduce the compensator dimension still further, consider the case where $z = \eta$. It is easy to see that the observability index for such a system will be 2. Thus, a functional observer of dimension 1 can be constructed [6]. A functional observer has the form

$$\begin{aligned} \dot{w} &= F w + G z + H u \\ w &= M w + P z \end{aligned} \quad (26)$$

and the property that w approaches $-kz$ exponentially, with time constants determined by the eigenvalues of F .

To implement the functional observer, two measurements of the form $G\eta$ and $P\dot{\eta}$ are needed, where G and P are $1 \times N$ vectors computed from the system parameters and the state feedback gain. The previously developed theory can be used to show how such measurements can be made on an Euler-Bernouli beam using two optical fiber sensors.

For an Euler-Bernouli beam model, the mode shapes, $\phi_i(x)$, are not only orthonormal, but satisfy the additional orthogonality relation

$$\int_0^l \phi_i''(x) \phi_j''(x) dx = \beta_i^4 \delta_{ij}. \quad (27)$$

Thus, the previous theory can be used to see that choosing

$$K(x) = \sum_{i=1}^N \frac{g_i}{\beta_i^4} \phi_i''(x), \quad (28)$$

where g_i is the i 'th element of G , makes the output of the fiber sensor equal to $G\eta$, as desired. By using two weighted fiber sensors a first-order compensator can be developed to implement the desired control law.

To illustrate these results, a four-mode model of an Euler-Bernouli beam was considered. The tip displacement and strain energy for an initial condition response of the open-loop system are shown in Figure 1. The response for a closed-loop system using a full-order compensator, where the measurements $L\eta$ were obtained using fiber sensors in a manner analogous to that described above for the functional observer, is shown in Figure 2. Figure 3 shows the response for a closed-loop system using a functional observer. The response is almost identical to that of a full-order compensator, even though the compensator order has been reduced from eight to one. The weighting functions for the fibers used with the functional observer are shown in Figure 4.

One problem usually encountered in using functional observers is that the direct feedthrough of the output results in excessive noise in the estimates. Fiber sensors, however, have a very high signal to noise ratio, and this should not be a problem. Also, as discussed in Section 4.3, spillover noise will not be a problem. The major tradeoff between a full-order compensator implemented using point measurements and a first-order compensator implemented using weighted fiber sensors is between the complexity of the compensator and the complexity of implementing the distributed gain in the fiber.

5. Conclusions

In this paper, we have shown that by filtering the output of a distributed sensor, we can produce a scalar output corresponding to any desired linear combination of the system position states. This result illustrates one of the advantages of distributed and distributed-effect sensors over point sensors - the set of scalar outputs realizable using point sensors is a subset of the set of outputs realizable using distributed sensors and filtering.

We also described three applications of distributed sensor filtering. Other applications that are currently under investigation include the use of distributed filtering for the placement of transfer function zeros, and the optimization of measures of observability.

Acknowledgements

This work was supported in part by NASA Grants NAG-1-1006, NAG-1-1043 and NTG-5004, ONR Grant N00014-89-J-3123, and ONR/DARPA contract N00014-88-0721.

References

- [1] S. S. Welch, "Optical distributed sensing and computation for a proposed flexible beam experiment," in *Proceedings of the Seventh VPI&SU Symposium on Dynamics and Control of Large Space Structures*, May 1989, Blacksburg, VA.
- [2] M. F. Barsky, K. M. Reichard, D. K. Lindner, and R. O. Claus, "Distributed methods for controlling flexible structures," in *Proceedings of the Seventh VPI&SU Symposium on Dynamics and Control of Large Space Structures*, May 1989, Blacksburg, VA.

- [3] M. F. Barsky and D. K. Lindner, "Distributed sensing for robust control of flexible structures," in Proceedings of the 1989 Allerton Controls Conference, September 1989, Urbana, IL.
- [4] Wie-Wen Chaing, and Chih-Kung Lee, "Critical damping control of a flexible slender plate using a distributed modal actuator and sensor," in Proceeding of the 1989 American Controls Conference, June 1989, Pittsburgh, PA.
- [5] M. J. Balas, "Feedback control of flexible systems," IEEE Transactions On Automatic Control, Vol. AC-23, No. 4, pp 673-679, August 1978.
- [6] C-T Chen, Linear System Theory and Design. Holt, Rinehart and Winston, 1984.

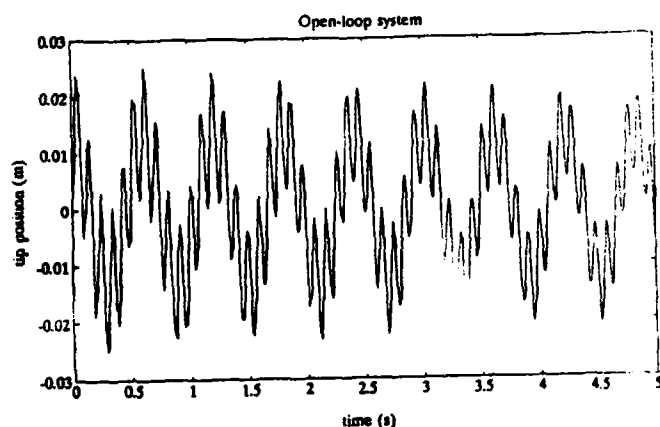


Figure 1.a Response of the open-loop system to initial conditions.

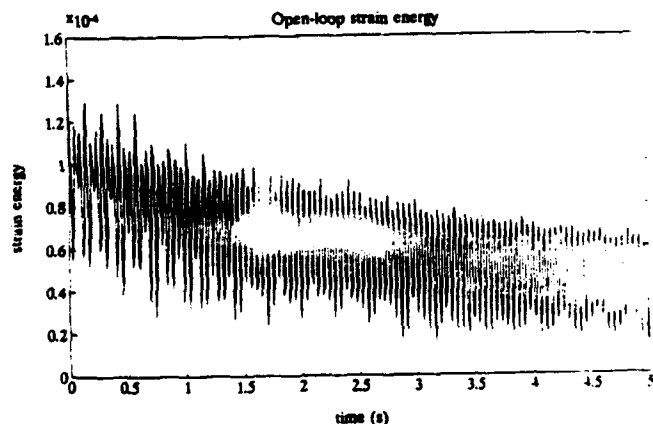


Figure 1.b Response of the open-loop system to initial conditions.

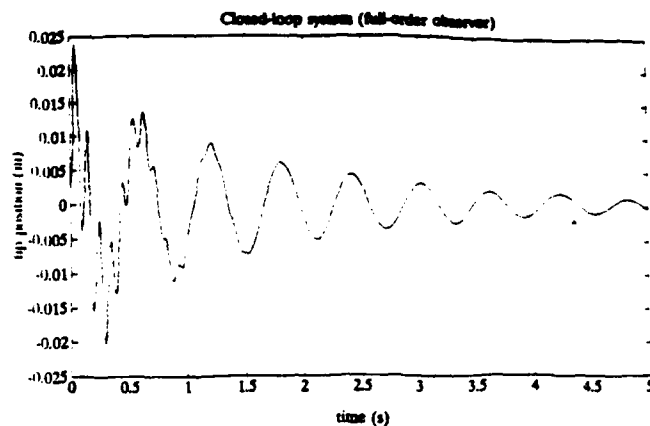


Figure 2.a Closed-loop system response using full-order observer.

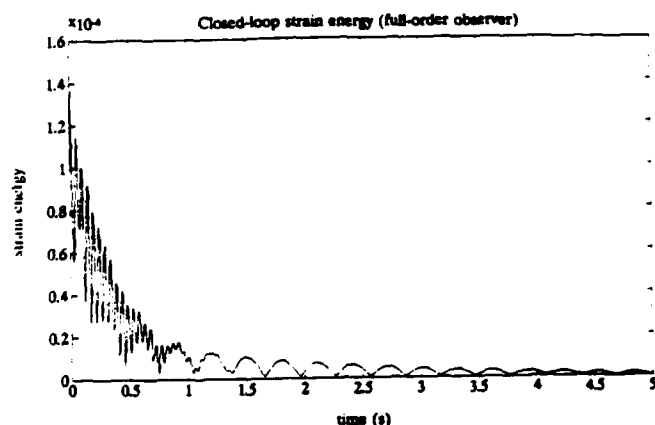


Figure 2.b Closed-loop system response using full-order observer.

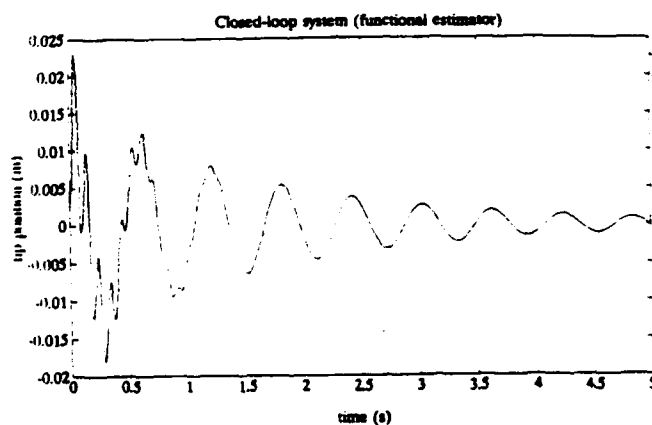


Figure 3.a Closed-loop system response using functional observer.

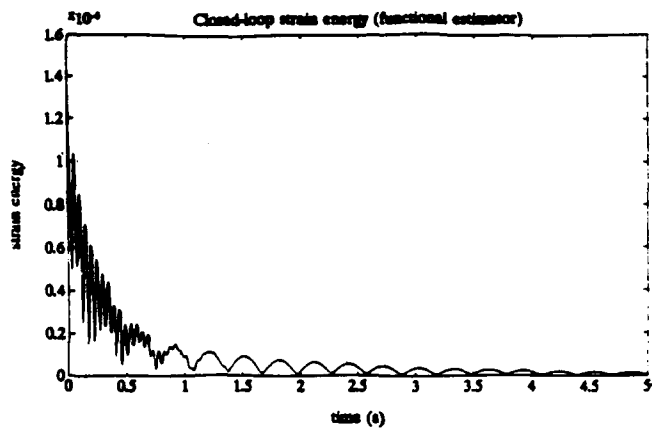


Figure 3.b Closed-loop system response using functional observer.

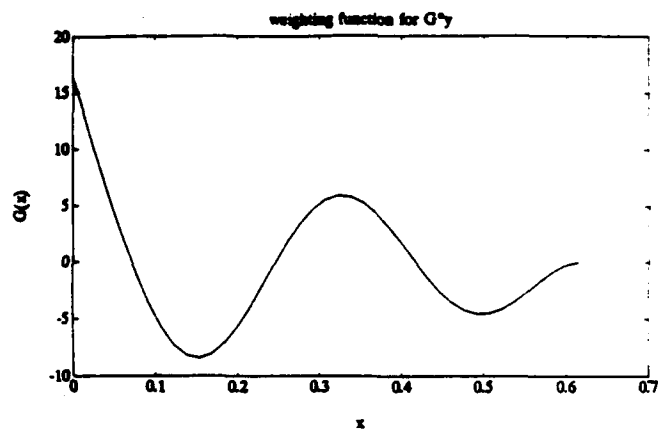


Figure 4.a Distributed filter weighting functions used to implement the functional observer.

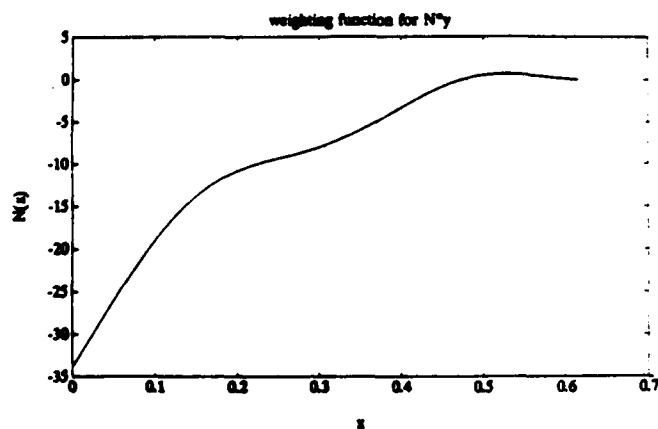


Figure 4.b Distributed filter weighting functions used to implement the functional observer.

APPLICATION OF FUZZY LOGIC CONTROL TO ACTIVE VIBRATION DAMPING

Apostolos Tsoukkas
and
Hugh VanLandingham

The Bradley Department of Electrical Engineering
Virginia Polytechnic Institute and State University
Blacksburg, Virginia 24061 - 0111

Abstract

In this paper the method of using Fuzzy Logic is explained and an application of Fuzzy Logic Control (FLC) to actively damp the vibrations of a cantilever beam is investigated. The results of this method are compared to controlling the structure using optimal control. The primary advantage of FLC is that satisfactory control can be achieved without detailed knowledge of the plant. A by-product of the method is that the FLC is able to utilize the maximum control effort available at the actuator.

Introduction

Since the early 1970's it has been proposed that the theory of "Fuzzy Logic" could be used to design control systems for poorly modeled and/or complex systems [1]. To date many successful applications have been documented [2,3,4]. The principal advantage of a Fuzzy Controller is its performance and simplicity of design in the absence of accurate plant models. The typical Fuzzy Controller is a combination of Fuzzy Logic and rule-based expert systems. The "rules" express the control policy much as people do. The fact that rule conditions are stated in "human" (qualitative) terms, such as "the error is moderately large", rather than in "computer" (quantitative) terms which require numerical ranges, converts the use of very few rules into an effective controller. Lee [5] provides an excellent tutorial paper with a comprehensive collection of references.

The past few years have witnessed an exponential growth in the applications of Fuzzy Control; however, the major interest has been in Japan and Europe and not in the U.S. where the concept originated. Outside of academic investigations the use of Fuzzy Control seems to have been relegated to household appliances, printers, and other "simple" controllers. One exception is that of the controllers used on some of the Japanese trains [6]. But in all cases the applications appear to be slight improvements in controllers which could have been designed by classical techniques.

A Fuzzy Logic Controller (FLC), see Fig. 1, can be viewed as a way of converting expert knowledge into an automatic control strategy without a detailed knowledge of the plant. The input to the FLC is fuzzified, or in other words, converted to the Fuzzy Set Domain. Then it is processed by the Fuzzy Inference Engine and a Fuzzy output is obtained (in the form of a sum of weighted Fuzzy Sets). That in turn is defuzzified, or converted back into a number in the real Domain which is used as the control input to the process to be controlled. To provide a benchmark for comparison, the example system will be controlled using an observer-based controller. With observer-based control (Fig. 2) the closed loop system consists of an observer to reproduce the states from the output and state variable feedback with gains derived, e.g. by optimal control, to provide the feedback correction.

TSOUKKAS AND VANLANDINGHAM

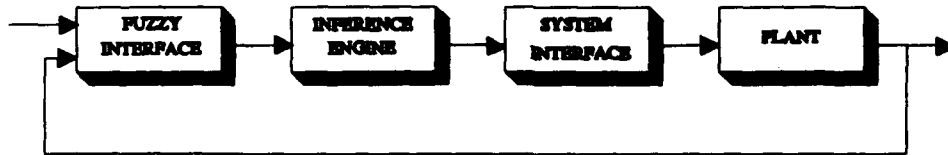


FIGURE 1. BLOCK DIAGRAM OF A FUZZY CONTROLLER

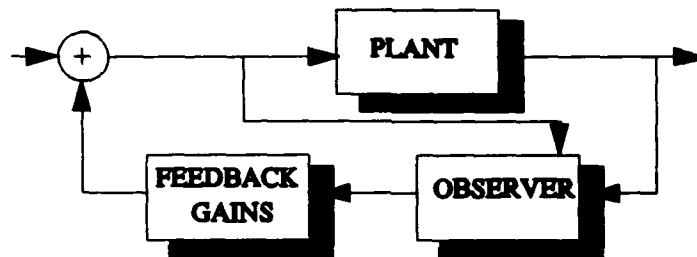


FIGURE 2. OBSERVER BASED CONTROL

Both of these ideas will be applied to actively damp the vibration of a cantilever beam. A system will be assumed for simulation purposes and both control strategies will be compared.

Fuzzy set theory

Fuzzy set theory was developed in an effort to deal with the uncertainty and impreciseness that is abundant in the real physical world. It was expected to model human reasoning, the ability of the human mind to deal with vague terms, and the ability to make decisions based on imprecise data.

The definitions on Fuzzy Set Theory are as they appeared on Zadeh's[8] seminal paper:

A Fuzzy Set is a class of objects with a continuum of grades of membership. It is characterized by a membership (or characteristic) function $\mu_A(x)$ which assigns to each object, a grade of membership ranging between 0 and 1. More formally:

Let X be a space of points and x in X (X is also called the Universe of discourse).

Then a Fuzzy set $A \in X$ is characterized by a membership function $\mu_A(x)$ which associates with each point in X a real number in the interval $[0,1]$ with the value of $\mu_A(x)$ at x representing the

APPLICATION OF FUZZY LOGIC CONTROL

"grade of membership" of x in A .

An ordinary set then is a set where $\mu_A(x)$ only takes the values of 0 and 1 on X .

An ordinary set could be viewed as a special case of a Fuzzy Set, or a Fuzzy Set could be viewed as an extension of an ordinary set.

Two fuzzy sets are equal if they have identical membership functions for all objects of X .

$$A = B \Leftrightarrow \mu_A(x) = \mu_B(x), \forall x \in X$$

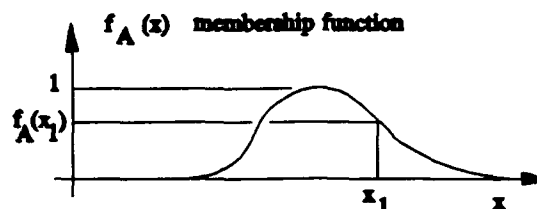


FIGURE 1. A GENERIC FUZZY SET

The complement of a fuzzy set A is denoted by A' and is defined by

$$\mu_{A'}(x) = 1 - \mu_A(x), \forall x \in X$$

A fuzzy set A is contained in B (or is a subset of B , or is smaller than or equal to B) if and only if the membership function of A is smaller than or equal to the membership function of B for all objects of X .

$$A \subseteq B \Leftrightarrow \mu_A(x) \leq \mu_B(x), \forall x \in X$$

Union: The union of two fuzzy sets A and B with membership functions $\mu_A(x)$ and $\mu_B(x)$ respectively is a fuzzy set C written as $C = A \cup B$ where the membership function of C at x is the maximum of the 2 membership functions.

$$\mu_C(x) = \max[\mu_A(x), \mu_B(x)], \forall x \in X$$

or in abbreviated form

$$\mu_C(x) = \mu_A(x) \vee \mu_B(x)$$

(The union of A and B is the smallest fuzzy set containing both A and B)

Intersection: The intersection of two fuzzy sets A and B with respective membership functions $\mu_A(x)$ and $\mu_B(x)$ is a fuzzy set C written as $C = A \cap B$ whose membership function is related to those of A and B by

$$\mu_C(x) = \min[\mu_A(x), \mu_B(x)], \forall x \in X$$

or in abbreviated form

$$\mu_C(x) = \mu_A(x) \wedge \mu_B(x)$$

(The intersection of A and B is the largest fuzzy set which is contained in both A and B).

The Algebraic product of two fuzzy sets A and B is defined by their membership functions as $\mu_{AB}(x) = \mu_A(x)\mu_B(x)$. (Note $AB \subseteq (A \cap B)$) and the Algebraic Sum is defined as $\mu_{A+B}(x) = \mu_A(x) + \mu_B(x)$ (Note that $((A \cup B) \subseteq (A + B))$).

A normal fuzzy set is a set that attains the maximum of 1.

TSOUKKAS AND VANLANDINGHAM

A convex fuzzy set is a set such that (see Fig. 3)

$$\mu_A[\lambda x_1 + (1-\lambda)x_2] \geq \min[\mu_A(x_1), \mu_A(x_2)], \quad x_1, x_2 \in X, \quad \lambda \in [0,1]$$

A fuzzy number is defined as the fuzzy set which is both normal and convex (Note that the set in Fig. 3 is a fuzzy number).

A fuzzy partition on the universe of discourse is the association of each linguistic variable with a term set on the universe of discourse (PB, PM, ...). So a linguistic variable takes on values of different fuzzy sets defined on a certain universe of discourse. An example of a fuzzy partition of the input space (of the fuzzy controller) is given in Appendix A Figure 1. You can note that the primary set of this partition is a triangular set.

The last definition we are interested in at this time is the concept of a fuzzy relation or fuzzy implication which is a generalization of the concept of a function. An n -ary fuzzy relation in X is a fuzzy set in the product space $X \times X \times \dots \times X$. For such relations the membership function is of the form $\mu_A(x_1, x_2, \dots, x_n)$ where $x_i \in X$, $i = 1, 2, \dots, n$. This definition is the one that permits the use of fuzzy logic control.

Fuzzy Logic Controller

Control was one of the first areas where Fuzzy Set Theory was applied and it has enjoyed a great deal of success in applications where the systems to be controlled were ill-defined, or where the process dynamics were more or less defined by the experience of an expert operator. Most of these systems have a very slow response but with the emergence of Fuzzy Logic chips the ability to control faster responding systems was realized.

The FLC provides an algorithm which can convert the linguistic control strategy based on expert knowledge into an automatic control strategy. The essential part of the FLC is a set of linguistic control rules related by the dual concepts of fuzzy implication and the compositional rules of inference (Lee[5]).

The FLC consists of 3 components (see Fig. 1):

- A. The Fuzzification interface
- B. The knowledge base and the decision making logic
- C. The defuzzification interface

A designer has to take into account many design variables that have to do with how each linguistic variable is defined and how you go from the Real domain to the Fuzzy Set domain and back.

A. The fuzzification interface

- 1) measures the values of the input variables (output and input process variables).
- 2) performs a scale mapping that transfers the range of values of input variables into corresponding universes of discourse.
- 3) converts the input data into suitable linguistic values which may be viewed as labels of fuzzy sets (or linguistic hedges, basically the labels PS, NM etc).

B. The knowledge base comprises a knowledge of the application domain and the attendant control goals.

- 1) data base provides necessary definitions
- 2) rule base characterizes the control goals and the control policy of the domain experts by means of a set of linguistic control rules.

The decision making logic is the heart of an FLC; it has the capability of simulating human decision making and of inferring fuzzy control actions, employing fuzzy implication and the rules of inference in the fuzzy logic.

C. The defuzzification interface performs

- 1) scale mapping which converts the output variables into corresponding universes

APPLICATION OF FUZZY LOGIC CONTROL

of discourse.

2) defuzzification which yields a nonfuzzy control action from an inferred fuzzy control rule.

The fuzzy control rules, also known as linguistic description rules have the format: "if (a set of conditions are satisfied) then (a set of consequences can be inferred)" and they are implemented by a fuzzy implication (a fuzzy relation).

Very important to the database is the normalization /discretization of the universes of discourse, the fuzzy partition of the input and the output spaces, the completeness of the data base and the choice of the function for the primary fuzzy set. For the rule base it is the choice of the process variables the source and derivation of the fuzzy control rules, the types of the fuzzy control rules, their consistency, interactivity and completeness. These are the guidelines a designer has to follow and take into account in order for the FLC to be effective.

A popular way for deriving a real output from the system (defuzzification) is the method of the centroid. Suppose that 2 or more rules apply in a particular case the j th applied rule being (Appendix A Fig. 2):

if x_1 is I_{1j} x_2 is I_{2j} ... , then y is B_{1j}

then the weight w_j is given by

$$w_j = I_{1j}(\tilde{x}_1) \wedge I_{2j}(\tilde{x}_2) \wedge \dots \wedge I_{nj}(\tilde{x}_n)$$

or for simplicity

$$w_j = I_{1j}(\tilde{x}_1) * I_{2j}(\tilde{x}_2) * \dots * I_{nj}(\tilde{x}_n)$$

Then respectively B^* (the Fuzzy output set) for both cases is

$$B^* = \bigcup_j w_j B_{1j} \quad \text{and} \quad B^* = \sum_j w_j B_{1j} .$$

The real output y can then be inferred by taking

$$y = \frac{\int B^*(y) y dy}{\int B^*(y) dy}$$

The first case is what is known as the max-min convolution, and the second is the sum-product convolution.

System

For simulation purposes the equations used were of a cantilever beam where both the actuator and the sensor were located at the tip of the beam. The first 3 modes were used. The system was then discretized and the discrete system was simulated. The sampling frequency was 5 KHz.

From VanLandingham [11] the equations of the continuous system were:

$$\begin{aligned} \dot{X} &= A X + B u \\ y &= C X \end{aligned}$$

In the next section we present some comparative results using FLC and a standard state feedback controller to damp the first mode of a three-mode system.

Simulation results

The input to the FLC is the output of the process and the change in the output (since setpoint = 0), that is:

$$\text{error} = y(k)$$

$$\text{change in error} = y(k) - y(k-1)$$

The output of the FLC is the control effort to be applied to the process and the scaling factor to the output is the maximum control effort that the actuator can supply. The scaling factors on the inputs are: the position of the peak of the PB term, the term set of the Fuzzy partition, and the input Universe of discourse (Appendix A. Fig 1).

The rule base used is as appears in Fig. 4 and is of the form: if y is "row _{i} " and ce is "column _{j} " then output is "element _{ij} ".

The defuzzification strategy used is the centroid method described above using sum-product convolution.

The results for the simulation of the controlled system using different scaling factors on the input and output spaces appear in the final four figures. The 3 numbers appearing at the bottom are the scaling factors on the error(y), the change of error, and the output(u).

As the output scaling factor is increased (the control effort applied), the system responds faster, however if it is increased too much the response does not settle to the set-point. If the input scaling factors (especially error) are decreased, finer control is obtained and the response is faster. The two scaling factors on the output and the input, could be tuned and then be reduced as the set-point is approached, thus continuously driving the system to the set point. These scaling factors could also be saved in the form of look-up tables, increasing the number of rules of the system but also increasing the adaptability of the controller.

For the optimal control simulation the assumption was that there was only information on the dynamics of the first mode (since FLC does not use any knowledge of the system dynamics). The response was never driven to the set point, but only oscillating around it Fig. 8). Also with time the oscillations grew due to the "spillover" effects.

Conclusion

The design of Fuzzy Logic Controllers was reviewed. The FLC is an inherent nonlinear control method which implements a set of linguistic (i.e. understandable to a human) rules. It typically performs much better in the absence of exact knowledge of the system dynamics and, additionally, can be tuned to give an optimal performance for a particular system. A comparison was made between the FLC and a standard optimal state feedback controller. The FLC is shown to have a clear advantage over the Optimal Controller because information on the system dynamics was restricted to one mode in the three-mode system.

| $y \backslash ce$ | PB | PM | PS | ZE | NS | NM | NB |
|-------------------|----|----|----|----|----|----|----|
| PB | NB | NB | NB | NB | NM | NS | ZE |
| PM | NB | NB | NB | NM | NS | ZE | PS |
| PS | NB | NB | NM | NS | ZE | PS | PM |
| ZE | NB | NM | NS | ZE | PS | PM | PB |
| NS | NM | NS | ZE | PS | PM | PB | PB |
| NM | NS | ZE | PS | PM | PB | PB | PB |
| NB | ZE | PS | PM | PB | PB | PB | PB |

FIGURE 4. Rule Base

APPLICATION OF FUZZY LOGIC CONTROL

FIGURE 5. Fuzzy Controller

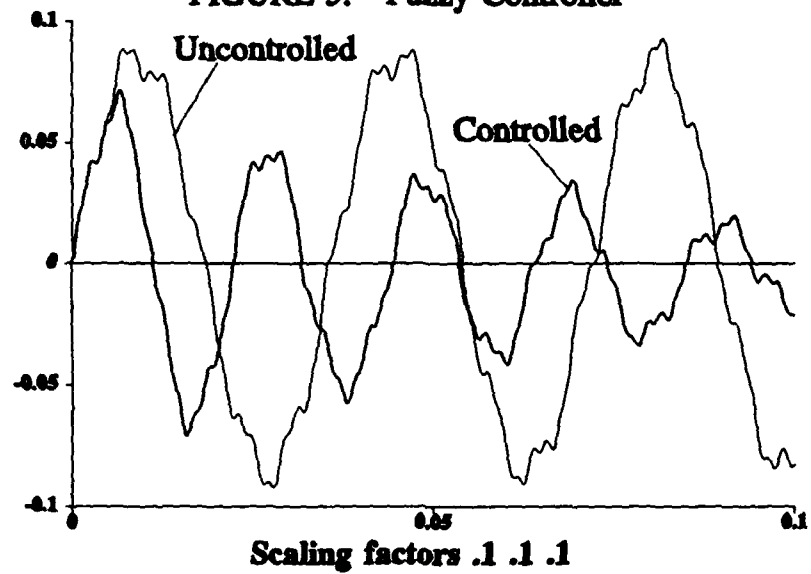


FIGURE 6. Fuzzy Control

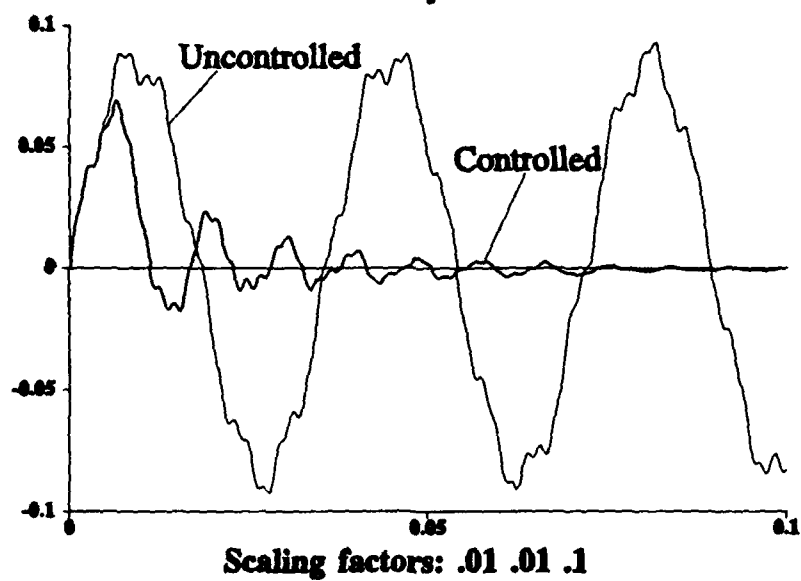


FIGURE 7. Fuzzy Control

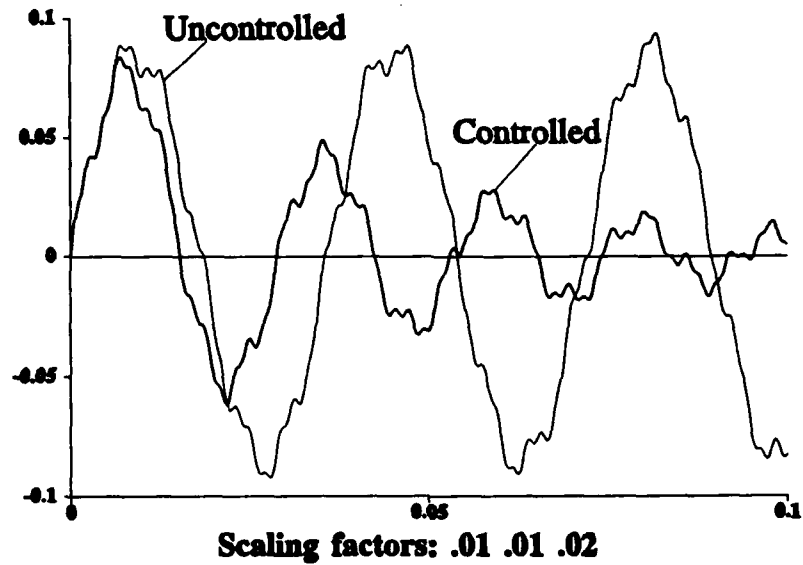
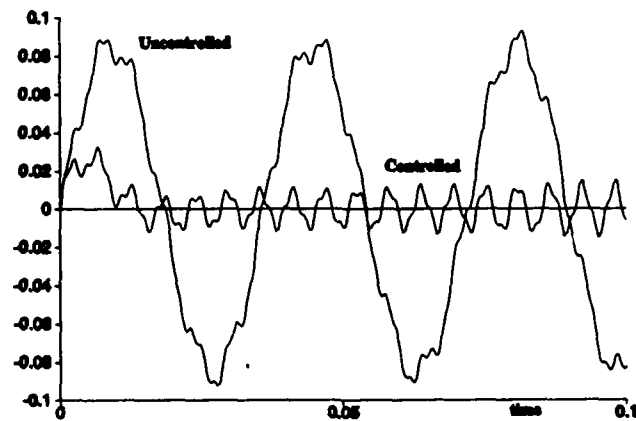


FIGURE 8.

Optimal Control Response



APPLICATION OF FUZZY LOGIC CONTROL

References:

1. Zadeh, L.A., "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes", *IEEE Transactions on Systems, Man and Cybernetics*, vol. SMC-3, no.1, pp. 28-44, Jan. 1973.
2. Mandič, N.J., Scharf, E.M., and Mamdani, E.H., "Practical Application of a Heuristic Fuzzy Rule-Based Controller to the Dynamic Control of a Robot Arm", *Proceedings of the IEE*, vol. 132, pt. D, no. 4, pp. 190-203, July 1985.
3. Larkin, L., "A Fuzzy Logic Controller for Aircraft Flight Control", *Industrial Applications of Fuzzy Control*, M. Sugeno, Editor. Amsterdam: Elsevier Science Publishers B. V. (North-Holland), 1985, pp. 87-103.
4. King, P.J., and Mamdani, E.H., "The Application of Fuzzy Control Systems to Industrial Processes", *Automatica*, vol. 13, no. 3, pp. 235-242, May 1977.
5. Lee, C.C., "Fuzzy Logic in Control Systems", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 20, no. 2, March 1990, pp. 404-435.
6. Yasunobu, S., Miyamoto, S., and Ihara, H., "Fuzzy Control for Automatic Train Operation System", *Proceedings of the 4th IFAC Conference: Control in Transportation Systems*, Baden-Baden, Germany, April 1983, pp. 33-39.
7. Gupta, M.M., Kiska, J.B., and Trojan, G.M., "Multivariable Structure of Fuzzy Control Systems", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-16, no. 5, September 1986, pp. 638- 656.
8. Zadeh, L.A., "Fuzzy sets", *Information Control*, Vol 8, 1965, pp 338-353.
9. Sugeno, M., "An Introductory Survey of Fuzzy Control", *Information Sciences*, vol 36, 1985, pp 59-83.
10. Li, Y.F., Lau, C.C., "Development of Fuzzy Algorithms for Servo Systems", *IEEE Control Systems Magazine*, April 1989, pp 65-71.
11. VanLandingham, H.F., Moose, R., "A study of the transverse vibration of a beam", (unpublished notes).

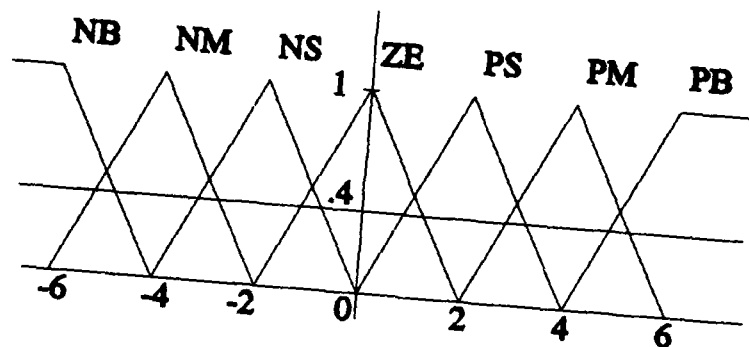


Figure A.1: Fuzzy Partition of the Universe of Discourse

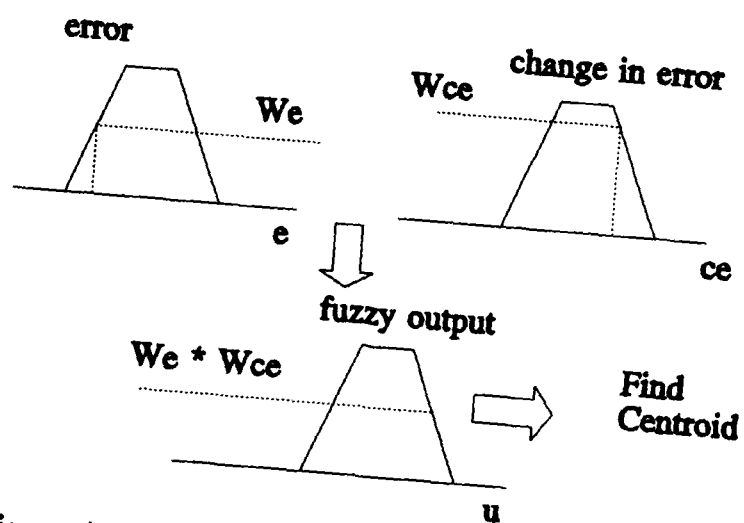


Figure A.2: Implementing a Fuzzy Relation

Deterministic Identification of Linear Multi-variable Systems.

Bhaskar. M. Gorti, S. Bingulac, H. F. VanLandingham

Bradley Department of Electrical Engineering.
Virginia Polytechnic Institute and State University.
Blacksburg, VA, 24061.

Abstract

This paper presents a simple computational algorithm for determining a state space model of a MIMO Discrete System from Input-Output Data. The obtained state space model is in a Pseudo-Observable Canonical Form. Unlike, all other deterministic identification procedures, the approach suggested in this paper, does not require structural identification [5], [6], i.e., determination of a unique set of observability indices. Instead, as it has been presented elsewhere [1], [2], a MIMO system could be represented in pseudo-observable canonical form, based on a set of admissible pseudo-observability indices [3].

Introduction

Consider, a linear time-invariant discrete system. It is known, [1]-[3], that based on a selected set of admissible pseudo observability indices:

$n = \{n_1, n_2, \dots, n_p\}$; p being the number of inputs, any

n^{th} order MIMO discrete system could be represented by the following pseudo-observable canonical form:

$$x(k+1) = A_o x(k) + B_o u(k) \quad (1)$$

$$y(k) = C_o x(k) + D_o u(k); \quad (2)$$

$$x(0) = x_0 \quad (3)$$

where $x \in R^n$, $u \in R^m$, $y \in R^p$, are the state, input and output vectors respectively, while A_o , B_o , C_o and D_o are matrices of compatible dimensions. In [3], it has also been shown that the total number of sets of admissible pseudo observability indices is less than

$$\text{or equal to } I = \frac{(n-1)!}{(p-1)!(n-p)!} \quad (4)$$

The pair (A_o, C_o) in the pseudo observable form is characterised by the following structure.

$$A_o = \begin{bmatrix} 0 & \dots & 0 & 1 & & & \\ 0 & & \dots & 0 & 1 & & \\ a_{11} & \dots & & a_{1j} & \dots & & a_{1n} \\ 0 & & & 0 & 1 & & \\ & & & & & \ddots & \\ 0 & & & & & \dots & 1 \\ a_{21} & \dots & & a_{2j} & \dots & & a_{2n} \\ 0 & & & 0 & & 0 & 0 & 1 \\ a_{p1} & \dots & & a_{pj} & \dots & & a_{pn} \end{bmatrix} \quad (5)$$

$$C_o = \begin{bmatrix} 1 & 0 & \dots & & \\ 0 & 1 & 0 & \dots & \\ 0 & 0 & 1 & 0 & \dots \end{bmatrix} \quad (6)$$

From (5) & (6), it can be concluded that A_o has only p rows with non-zero and non-unity elements. Location of these rows $s = \{s_1, s_2, \dots, s_p\}$ are uniquely determined by the set

of assumed pseudo observability indices n_i . The remaining $(n-p)$ rows of A_o correspond to the last $(n-p)$ rows of the Identity matrix I_n . The first p rows of this Identity matrix correspond to the rows in C_o . Matrices B_o and D_o do not have any specific structure.

$$B_o = \begin{bmatrix} b_{11} & \dots & b_{1m} \\ & \dots & \\ & & \vdots \\ & & b_{n1} & \dots & b_{nm} \end{bmatrix} \quad (7)$$

$$D_o = \begin{bmatrix} d_{11} & \dots & d_{1m} \\ & \dots & \\ & & \vdots \\ d_{p1} & \dots & d_{pm} \end{bmatrix} \quad (8)$$

Identification Identity

It is assumed, that the Input-Output sequences:

$$\{u(k), y(k)\}; \quad k = 0, 1, 2, \dots, N-1;$$

$$u(k) = u_k; y(k) = y_k;$$

corresponding to an n^{th} order system are available. The approach suggested in this paper, determines a system representation satisfying (1) - (3), where A_o , B_o , C_o and D_o are given by (5) - (8). In order to determine non-zero and non-unity parameters in A_o , B_o , C_o , D_o and x_0 the following procedure is suggested.

From (1) - (3), the following equation is obtained:

$$Y(k) = Q_o x(k) + H U(k) \quad (9)$$

where in (9) the matrices $Y(k)$ and $U(k)$ are $(l+1)p$ and $(l+1)p$ column vectors, while Q_o and H are matrices of the dimensions $(l+1)p \times n$, $n \times 1$ and $(l+1)p \times (l+1)$ respectively. Q_o is

This work is supported by the Office of Naval Research under Grant N-00014-89-J-3123.

the observability matrix of the pair (A_o, C_o) and $l = n-p+1$. The structure of these matrices is as follows:

$$U(k) = \begin{bmatrix} u_k \\ \vdots \\ u_{k+i} \\ \vdots \\ u_{k+l} \end{bmatrix} \quad Y(k) = \begin{bmatrix} y_k \\ \vdots \\ y_{k+i} \\ \vdots \\ y_{k+l} \end{bmatrix} \quad Q_o = \begin{bmatrix} C_o \\ C_o A_o \\ \vdots \\ C_o A_o^{l-1} \end{bmatrix} \quad (10)$$

$$H = \begin{bmatrix} D_o & 0 & \dots & 0 \\ C_o B_o & D_o & \dots & 0 \\ C_o A_o B_o & C_o B_o & \ddots & 0 \\ \vdots & \vdots & \ddots & 0 \\ & & & 0 & 0 \\ & & & D_o & 0 \\ C_o A_o^{l-1} B_o & \dots & C_o B_o & D_o \end{bmatrix} \quad (11)$$

Elements of vectors y_{k+l} , z_k and u_{k+l} are related to the elements in A_o , B_o , C_o and D_o by the set of $(l+1)p$ scalar equations given by (9). It could be verified that among $(l+1)p$ rows of the observability matrix Q_o , there are n rows equal to n rows of I_n and p rows corresponding to the non-zero, non-unity rows from A_o . The locations of the n rows corresponding to I_n and the p rows corresponding to A_o are uniquely determined by the assumed set of admissible pseudo-observability indices.

Now we select from the vector $Y(k)$ subvectors Y_{1k} and Y_{2k} corresponding to the n rows in Q_o containing rows equal to I_n and the p rows in Q_o containing non-zero, non-unity rows from A_o . Let the corresponding rows from H be H_1 and H_2 . Then, from equation (9) the following equations can be formed.

$$Y_{1k} = I z_k + H_1 U_k \quad (12)$$

$$Y_{2k} = A' z_k + H_2 U_k \quad (13)$$

$$\text{where } A' = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{p1} & \dots & a_{pn} \end{bmatrix} \quad (14)$$

Therefore, eliminating the state vector z_k from (12) and (13) the following Identification Identity is obtained;

$$Y_{2k} = \begin{bmatrix} B' & \vdots & A' \end{bmatrix} \begin{bmatrix} U_{1k} \\ \vdots \\ Y_{1k} \end{bmatrix} \quad (15)$$

where U_{1k} contains the first $m(n'+1)$ rows from U_k , where

$$n' = \max \{ n_i \}, \text{ for } i = 1, \dots, p \text{ and } B' = H_2 - A' H_1.$$

The matrix $\begin{bmatrix} B' & \vdots & A' \end{bmatrix}$ will be called the

"parameter matrix", since it is obvious that it depends only on the elements of A_o , B_o , C_o and D_o . Using a set of q available measurements, where q satisfies the condition, $n + (n'+1)m \leq q$, we define:

$$Y = \begin{bmatrix} Y_{2k} & Y_{2(k+1)} & \dots & Y_{2(k+q-1)} \end{bmatrix} \quad (16)$$

$$Z = \begin{bmatrix} U_{1k} & \dots & U_{1(k+q-1)} \\ \vdots & \ddots & \vdots \\ Y_{1k} & \dots & Y_{1(k+q-1)} \end{bmatrix} \quad (17)$$

The dimensions of Y and Z are $p \times q$ and $((n'+1)m+n) \times q$, respectively. Using the Least Squares Method [7], from (15) - (17), the parameter matrix can be expressed as:

$$\begin{bmatrix} B' & \vdots & A' \end{bmatrix} = Y Z^T (Z Z^T)^{-1} \quad (18)$$

where, in the case of a sufficiently rich input signal $u(k)$ and admissible set of pseudo observability indices, Z is a full row rank matrix. Since, we have A' , the matrix A_o is determined directly from (18) and (5), while C_o is known to have the structure as in (6).

Determination of B_o

It can be easily shown that $B_o = Q_c^e B^*$

$$\text{where } Q_c^e = \begin{bmatrix} B_e & A_o B_e & \dots & A_o^{n'} B_e \end{bmatrix}$$

The columns b_{e_j} of $(n \times p)$ "equivalent input matrix", B_e ,

contain $(n-1)$ zeros and only one unity, whose location is

determined by the integers j_i of the set j , where A_o has non-zero,

non-unity rows. B^* is formed by the following partitioning of B' .

$$B' = \begin{bmatrix} B'_0 & B'_1 & \dots & B'_{n'} \end{bmatrix} \quad (19)$$

$$B^* = \begin{bmatrix} B' \\ \vdots \\ B'_{n'} \end{bmatrix} \quad (20)$$

Note that the dimensions of B' , B^* and B'_i , $i = 0, \dots, n'$, are $p \times m(n'+1)$, $p(n'+1) \times m$ and $p \times m$, respectively.

Determination of D_o

From the definition of transfer function matrix we have:

$$G(z) = F^{-1}(z) N(z) = C_o (Iz - A_o)^{-1} B_o + D_o \quad (21)$$

$$D_o = F^{-1}(z) D(z) \text{ or } C_o (Iz - A_o)^{-1} B_o \quad (22)$$

where $N(z)$ and $F(z)$ are $(p \times m)$ and $(p \times p)$ co-prime polynomial matrices [4], where

$$N(z) = \sum_{i=0}^{n'} N_i z^i \text{ and } F(z) = \sum_{i=0}^{n'} F_i z^i. \quad \text{Since } D_o \text{ on the}$$

l.h.s of (22) does not depend on the Z-transform variable z, it could be calculated by evaluating the r.h.s of (22) for an arbitrary value of z. Since a discrete time system has no eigenvalue at z = 0, it follows that F_o and A_o are always non-singular. From [4] it can be concluded that N_o = B_o', (19) and F_o is given by the first p columns of A', leading finally to:

$$D_o = F_o^{-1} N_o + C_o A_o^{-1} B_o \quad (23)$$

Determination of Initial Condition x_o

The initial condition vector x_o, corresponding to (1) - (3), could be calculated directly from (12), by setting k = 0 i.e

$$x_o = Y_{1o} - H_1 U_o \quad (24)$$

Selector Vector Algorithm

1. Define a Set $\eta \equiv \{n_1, \dots, n_p\}$, of admissible pseudo observability indices where p is the number of outputs of the system.
2. Set $n = \sum_{i=1}^p n_i$; where n is the order of the system,
- $n' = \max \{n_i\}$ and $n_p = (n' + 1) * p$.
3. Set i = 1.
4. Set h = i.
5. For j = 1 through n_p, Set $\hat{V}_x(h) = n_i + 1 - j$, h = h + p.
6. Set i = i + 1.
7. If i > p, go to 8; else, go to 4.
8. Set i_1 = 0, i_2 = 0, k = 1.
9. If $\hat{V}_x(k) < 0$, go to 11; else, go to 10.
10. Set i_1 = i_1 + 1, h(i_1) = k.
11. If $\hat{V}_x(k) \neq 0$, go to 13; else, go to 12.
12. Set i_2 = i_2 + 1, r(i_2) = k.
13. Set k = k + 1.
14. If k ≤ n_p, go to 9; else, go to 15.
15. Set q = p + 1, i_a = 0, i_p = 0, i_i = 0.
16. Set i_p = i_p + 1.
17. If $\hat{V}_x(q) < 0$, Set i_p = i_p - 1 and go to 21; else, go to 18.
18. If $\hat{V}_x(q) \neq 0$, Set i_i = i_i + 1; else, go to 20.
19. Set s_c(i_i) = i_p.
20. Set i_a = i_a + 1, s(i_a) = i_p, q = q + 1.
21. Set q = q + 1.
22. If q ≤ n_p, go to 16; else, Stop.

Example

For p = 3 and $\eta = \{1, 4, 2\}$, the above suggested

algorithm gives the following selector vectors:

$$s = \{1, 5, 7\}, s_c = \{2, 3, 4, 6\}, \\ h = \{1, 2, 3, 5, 6, 8, 11\} \text{ and } r = \{4, 9, 14\}.$$

Identification Algorithm

1. Given an Input-Output sequences $\{u(k), y(k)\}$, k = 0, 1, ..., N-1, corresponding to

an n^{th} order Linear MIMO discrete system.
 $u(k) \in R^m$; $y(k) \in R^p$
 where $1 \leq m \leq n$ and $1 \leq p \leq n$.

(25)

2. Assume a set $\eta = \{n_1^j, n_2^j, \dots, n_p^j\}$,

of admissible pseudo observability indices which satisfy the

$$\text{condition } n = \sum_{j=1}^p n_i^j; 1 \leq n_i^j \leq n - p + 1. \quad (26)$$

$$\text{Let } n' \triangleq \max_{j=1}^p \{n_1^j, n_2^j, \dots, n_p^j\}.$$

Build the $(n' + 1) \times N_1$ matrix U, where

$N_1 \leq n - (n' + 1)$, defined by;

$$U = \begin{bmatrix} U(0) & \dots & U(N_1-1) \\ U(1) & \dots & U(N_1) \\ \vdots & \vdots & \vdots \\ U(n') & \dots & U(N_1+n'-1) \end{bmatrix} \quad (27)$$

If the matrix U is of the full row rank, i.e

$$\rho(U) = (n' + 1)m, \quad (28)$$

then the input sequence is "sufficiently rich" and is capable of exciting all the n modes of the system to be identified.

If $\rho(U) < (n' + 1)m$, then either:

(a) Select another set of pseudo observability indices having a smaller value of n'

or

(b) Select a different, more rich input sequence which would satisfy the richness condition (28).

3. Using the selector vector algorithm determine the following:
 (a) A set of integres,

$$s^j = \{s_1^j, \dots, s_p^j\} \quad (29)$$

corresponding to the locations of non-zero, non-unity rows

$$a_i = [a_{i1}, \dots, a_{in}] \quad (30)$$

of the matrix A_o of the pseudo observable canonical form.

(b) A set of (n - p) integres s_c^j (complement to the

set s^j) corresponding to the locations of the last (n - p) rows from the Identity matrix I_n in the matrix A_o.

(c) A set of n integres:

$$h^j = \{h_1^j, \dots, h_n^j\} \quad (31)$$

corresponding to the location of the n rows of I_n in the observability matrix:

$$Q_o = \begin{bmatrix} C_o \\ \vdots \\ C_o A_o \\ \vdots \\ C_o A_o^{n-p} \end{bmatrix} \quad (32)$$

(d) A set of p integres;

$$r^j = \{r_1^j, \dots, r_p^j\}, \quad (33)$$

corresponding to the locations of the rows a_j in the observability matrix Q_o.

4. Build the p(n' + 1) × N_1 matrix Y, defined by;

$$Y = \begin{bmatrix} Y(0) & \dots & Y(N_1-1) \\ \vdots & \vdots & \vdots \\ Y(n') & \dots & Y(N_1+n'-1) \end{bmatrix} \quad (34)$$

5. From the matrix Y select the $(n \times N_1)$ matrix Y_1 with n rows corresponding to the elements in h^j . This selection could be represented by the following premultiplication;

$$Y_1 = S_h \times Y \quad (35)$$

where S_h is the "selector matrix" of dimension $(n \times N_1)$.

The k^{th} row of S_h have $(N_1 - 1)$ zeros and a unity at the location specified by the integers h_k^j of the set h^j .

6. From the matrix Y select the $(p \times N_1)$ matrix Y_2 with p rows corresponding to the elements in r^j . This selection could be represented by the following premultiplication;

$$Y_2 = S_r \times Y \quad (36)$$

where S_r is the "selector matrix" of dimension $(p \times N_1)$.

The k^{th} row of S_r have $(N_1 - 1)$ zeros and a unity at the location specified by the integers r_k^j of the set r^j .

7. Build the matrix Z ; $Z \triangleq \begin{bmatrix} U \\ \dots \\ Y_1 \end{bmatrix}$; (37)

If the matrix Z is of full row rank, then the selected set of pseudo observability indices are admissible.

8. Using the Least Squares algorithm determine the $p \times [(n' + 1)m + n]$ "parameter matrix" Q ,
where $Q = [B' : A']$, satisfying

$$Y_2 = Q \times Z = [B' : A'] \begin{bmatrix} U \\ \dots \\ Y_1 \end{bmatrix} \quad (38)$$

9. B' and A' are $p \times (n' + 1)m$ and $(p \times n)$ matrices, respectively. The p rows in the matrix A' correspond to the non-zero, non-unity rows, (30), of the matrix A_0 . Thus having determined the parameter matrix Q , i.e the matrix A' ,

$$A' = \begin{bmatrix} a_1 \\ \dots \\ \vdots \\ \dots \\ a_p \end{bmatrix} \quad (39)$$

The matrix A_0 in the pseudo observable canonical

form can be easily determined using the sets g^j and g_c^j .

10. As it has been mentioned earlier, the matrix C_0 in the pseudo observable canonical form is always of the structure is,

$C_0 = [I_p : 0]$, i.e it contains the first p rows from the

identity matrix I_n . The input matrix B_0 is to be determined by the following procedure.

11. The $p \times (n' + 1)m$ matrix B' is partitioned into $n' + 1$,

$(p \times m)$ submatrices $\tilde{B}_0, \dots, \tilde{B}_{n'}$ i.e:

$$B' = [\tilde{B}_0 : \dots : \tilde{B}_{n'}] \quad (40)$$

12. Let the columns b_{ej} of the $(n \times p)$ matrix B_e , "equivalent input matrix", contain $(n - 1)$ zeros and only one unity, whose location is determined by the integer s_k^j of the set g^j .

13. Build the $[n \times (n' + 1)p]$ controllability matrix Q_c of the pair $\{A_0, B_e\}$:

$$Q_c = [B_e : A_0 B_e : \dots : A_0^{n'} B_e] \quad (41)$$

14. The input matrix B_0 of the pseudo observable canonical form can now be easily determined from $B_0 = Q_c B^*$ (42)

$$\text{where } B^* = \begin{bmatrix} B'_0 \\ \dots \\ \vdots \\ \dots \\ B'_{n'} \end{bmatrix} \quad (43)$$

The "direct path" feed-through matrix D_0 in the pseudo observable canonical form is determined in the following way.

15. We know that,
 $G(z) = C_0 (Iz - A_0)^{-1} B_0 + D_0 = D^{-1}(z) N(z)$ for all z ,
or (44)

$$D_0 = D^{-1}(z) N(z) - C_0 (Iz - A_0)^{-1} B_0.$$

Since the matrix D_0 does not depend on z , D_0 can be determined by evaluating the above equation at $z = 0$, i.e

$$D_0 = D^{-1}(0) N(0) + C_0 A_0^{-1} B_0. \quad (45)$$

16. From the identification identity it can be seen that $N(0) = B'_0$ and $D(0) = \tilde{A}_1$, where $(p \times p)$ matrix \tilde{A}_1 , contains the first p columns from A' , i.e:

$$A' = [\tilde{A}_1 : \tilde{A}_2] \quad (46)$$

17. The initial condition vector $x(0) = x_0$ can be determined by the following equation:

$$\hat{y}(0) = x_0 - H_1 \hat{u}(0) \quad (47)$$

where $\hat{y}(0)$ is the first column of the matrix Y_1 and $\hat{u}(0)$ is the first column of the matrix U while H_1 is formed as follows;

$$H_1 = S_h \times H \quad (48)$$

where the $(n \times N_1)$ selector matrix S_h is given by (35).

Illustrative Example

In order to generate the input/output sequence $\{u(k), y(k)\}$ which will be used in the identification algorithm the following example is considered.

$$\begin{aligned} \dot{x} &= A x + B u \\ y &= C x + D u, \quad x(0) = x_0 \end{aligned} \quad (49)$$

where

$$A = \begin{bmatrix} .1 & 0 & 0 & 0 & 0 \\ 0 & .2 & 0 & 0 & 0 \\ 0 & 0 & .3 & 0 & 0 \\ 0 & 0 & 0 & .4 & 0 \\ 0 & 0 & 0 & 0 & .5 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 1 \\ .01 & 1 \\ .02 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \quad x_0 = \begin{bmatrix} 1 \\ 2 \\ 1 \\ 2 \\ 1 \end{bmatrix} \quad (50)$$

$$C = \begin{bmatrix} 1.01 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.01 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad D = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \end{bmatrix}$$

i.e. $n=5$, $m=2$ and $p=3$. For the input sequence $u(k)$, the following $m \times N$, $N = 17$, matrix consisting of pseudo random numbers is selected.

$$u(k) = \begin{bmatrix} 0.448 & 2.223 & 1.885 & 1.900 & 0.380 & 2.349 & 0.827 & -0.415 & 0.654 \\ 1.817 & 1.792 & -0.279 & 0.382 & 0.776 & -0.391 & 2.266 & -0.25 & 0.091 \\ -0.423 & 0.782 & -0.270 & -0.064 & 1.182 & 1.151 & 0.907 & 0.616 \\ -0.249 & 2.489 & 1.394 & 1.922 & 2.371 & 2.360 & 1.876 & 2.172 \end{bmatrix}$$

The output sequence $y(k)$ corresponding to (49) and (50) is:

$$y(k) = \begin{bmatrix} 1.020 & 1.491 & 4.184 & 2.025 & 2.488 & 1.413 & 2.096 & 3.326 \\ 0.572 & 4.357 & 4.382 & 2.408 & 0.957 & 3.313 & 0.769 & 1.873 \\ 7.000 & 10.722 & 14.675 & 4.768 & 5.197 & 5.556 & 2.122 & 12.472 \\ -0.105 & 0.736 & -0.601 & 3.235 & 1.464 & 2.026 & 3.782 & 3.916 & 3.197 \\ 1.309 & -0.120 & 0.613 & 2.084 & 2.209 & 3.769 & 4.349 & 4.277 & 3.543 \\ 3.012 & 2.335 & -0.965 & 13.112 & 10.489 & 12.986 & 17.277 & 18.285 & 15.977 \end{bmatrix}$$

According to [3], it follows that in the case of $n = 5$ and $p = 3$, the total number of possible sets of pseudo observable indices is equal to

$$I = \frac{(5-1)!}{(3-1)!(5-3)!} = 6. \text{ These 6 sets of indices are } \{1,1,3\},$$

$\{1,2,2\}$, $\{2,1,2\}$, $\{2,2,1\}$, $\{1,3,1\}$ and $\{3,1,1\}$. In case of (50), it can be concluded that the sets $\{3,1,1\}$ and $\{1,3,1\}$ are non-admissible and that the set $\{2,2,1\}$ corresponds to the unique set of pseudo observability indices [3],[4] and [6].

Using the available input/output sequence and applying the above mentioned algorithm for all the admissible sets of pseudo observable indices, we get:

Case I: $n^1 = \{1,1,3\}$. Using n^1 , the selector vectors algorithm gives;

$$s^1 = \{1,2,5\}, s_c^1 = \{3,4\}, h^1 = \{1,2,3,6,9\} \text{ \& } r^1 = \{4,5,12\}.$$

After building the identification identity (15) & (18), the following parameter matrix Q is obtained as in (38), where B^1 and A^1 are as given below;

$$A^1/T = \begin{bmatrix} .098 & .000 & .003 & -.015 & .017 \\ .002 & .299 & -.005 & .036 & -.051 \\ -.012 & .002 & .041 & -.383 & 1.103 \end{bmatrix}$$

$$B^1/T = \begin{bmatrix} 1.014 & 1.061 & -.017 & -.084 & .000 & .000 \\ -.310 & .908 & 1.052 & .254 & .000 & .000 \\ .285 & .808 & -1.028 & -4.015 & 1.030 & 5.000 \end{bmatrix}$$

Thus the system matrix A_0 in the pseudo observable canonical form is;

$$A_0 = \begin{bmatrix} .098 & .000 & .003 & -.015 & .017 \\ .002 & .299 & -.005 & .036 & -.051 \\ .000 & .000 & .000 & 1.000 & .000 \\ .000 & .000 & .000 & .000 & 1.000 \\ -.012 & .002 & .041 & -.383 & 1.103 \end{bmatrix}$$

The equivalent input matrix B_e used in calculating B_0 is:

$$B_e = \begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \text{ leading to } B_0 = \begin{bmatrix} 1.000 & 1.010 \\ 0.020 & 1.010 \\ 1.030 & 5.000 \\ 0.108 & 1.500 \\ 0.012 & 0.550 \end{bmatrix}$$

The initial condition vector is calculated by using step (17) of the algorithm;

$$x_0 = \begin{bmatrix} 1.020 \\ 1.020 \\ 7.000 \\ 2.100 \\ 0.750 \end{bmatrix}$$

The same procedure is followed for the other three sets of admissible pseudo observability indices and following results are obtained:

$$\text{Case II: } n^2 = \{1,2,2\}, s^2 = \{1,4,5\}, s_c^2 = \{2,3\},$$

$$h^2 = \{1,2,3,5,6\} \text{ \& } r^2 = \{4,8,9\}.$$

$$A^2/T = \begin{bmatrix} .099 & .099 & .002 & -.331 & -.003 \\ .000 & -.120 & .000 & .700 & .000 \\ .041 & 5.881 & -.101 & -19.7 & .701 \end{bmatrix}$$

$$B^2/T = \begin{bmatrix} .912 & 1.361 & .331 & .000 & .000 & .000 \\ .112 & -.403 & -.680 & 1.010 & 1.000 & .000 \\ -6.102 & 17.857 & 20.698 & 5.000 & .000 & .000 \end{bmatrix}$$

$$A_0 = \begin{bmatrix} .099 & .099 & .002 & -.331 & -.003 \\ .000 & .000 & .000 & 1.000 & .000 \\ .000 & .000 & .000 & .000 & 1.000 \\ .000 & -.120 & .000 & .700 & .000 \\ .041 & 5.881 & -.101 & -19.7 & .701 \end{bmatrix}$$

$$B_e = \begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \quad B_0 = \begin{bmatrix} 1.000 & 1.010 \\ 0.020 & 1.010 \\ 1.030 & 5.000 \\ 0.006 & 0.304 \\ 0.108 & 1.500 \end{bmatrix}$$

$$x_0 = \begin{bmatrix} 1.020 \\ 1.020 \\ 7.000 \\ 0.308 \\ 2.100 \end{bmatrix}$$

$$\text{Case III: } n^3 = \{2,1,2\}.$$

$$s^3 = \{2,4,5\}, s_c^3 = \{1,3\}, h^3 = \{1,2,3,4,6\} \text{ \& } r^3 = \{5,7,9\}.$$

$$A^3/T = \begin{bmatrix} .298 & .298 & .005 & -3.02 & -.010 \\ -.020 & .000 & .000 & .300 & .000 \\ -5.82 & .020 & -.201 & -59.4 & .902 \end{bmatrix}$$

$$B^3/T = \begin{bmatrix} 2.753 & 4.112 & 1.000 & .000 & .000 & .000 \\ -.200 & -.201 & 1.000 & 1.010 & .000 & .000 \\ -60.3 & -63.0 & 1.030 & 5.010 & .000 & .000 \end{bmatrix}$$

$$A_0 = \begin{bmatrix} .000 & .000 & .000 & 1.000 & .000 \\ .298 & .298 & .005 & -3.02 & -.010 \\ .000 & .000 & .000 & .000 & 1.000 \\ -.020 & .000 & .000 & .300 & .000 \\ -5.82 & .020 & -.201 & -59.4 & .902 \end{bmatrix}$$

$$B_e = \begin{bmatrix} 0.0 & 0.0 & 0.0 \\ 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \quad B_0 = \begin{bmatrix} 1.000 & 1.010 \\ 0.020 & 1.010 \\ 1.030 & 5.000 \\ 0.100 & 0.102 \\ 0.108 & 1.500 \end{bmatrix}$$

$$x_0 = \begin{bmatrix} 1.020 \\ 1.020 \\ 7.000 \\ 0.104 \\ 2.100 \end{bmatrix}$$

Case IV : $\eta^4 = \{ 2, 2, 1 \}$.

$\varepsilon^4 = \{ 3, 4, 5 \}$, $\varepsilon_c^4 = \{ 1, 2 \}$, $h^4 = \{ 1, 2, 3, 4, 5 \}$ & $r^4 = \{ 6, 7, 8 \}$.

$$A^4/T = \begin{bmatrix} 29.2 & 29.2 & .5 & -296.0 & -98 \\ -.020 & .000 & .000 & .300 & .000 \\ .000 & -.120 & .000 & .000 & .700 \end{bmatrix}$$

$$B^4/T = \begin{bmatrix} 269.8 & 402.9 & 97.9 & .000 & .000 & .000 \\ -.200 & -.201 & 1.000 & 1.010 & .000 & .000 \\ .112 & -.403 & -.680 & 1.010 & 1.000 & .000 \end{bmatrix}$$

$$A_0 = \begin{bmatrix} .000 & .000 & .000 & 1.000 & .000 \\ .000 & .000 & .000 & .000 & 1.000 \\ 29.2 & 29.2 & .500 & -296 & -98 \\ -.020 & .000 & .000 & .300 & .000 \\ .000 & -.120 & .000 & .000 & .700 \end{bmatrix}$$

$$B_e = \begin{bmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 1.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \quad B_0 = \begin{bmatrix} 1.000 & 1.010 \\ 0.020 & 1.010 \\ 1.029 & 5.000 \\ 0.100 & 0.102 \\ 0.006 & 0.304 \end{bmatrix}$$

$$x_0 = \begin{bmatrix} 1.020 \\ 1.020 \\ 7.000 \\ 0.104 \\ 0.304 \end{bmatrix}$$

Conclusion

The main purpose of this paper is to show that in identifying MIMO systems, the number of possible state space representations corresponds to the number of admissible sets of pseudo observable indices and that one of these sets, not necessarily equal to the unique set of pseudo observable indices, leads to the most convenient representation involving manipulation of well conditioned matrices. In this case the matrices A_0 and B_0 have elements with relatively small absolute values, which can be verified by comparing the Case I with the other three cases in the above illustrative example.

References

[1]. Gevers, M., and V. Wertz (1982). Uniquely identifiable state-space and ARMA parametrizations for multivariable linear systems, Automatica, 20, pp. 333-347.

[2]. Bingulac, S., and Krtolica, R. V. (1988). An Algorithm for Simultaneous order and parameter identification in Multivariable systems, IFACS - IFORS Symposium, Beijing, PRC, 27.

[3]. Bingulac, S., and Krtolica, R. V. (1987). On admissibility of pseudoobservability and pseudocontrollability indices, IEEE Trans. Aut. Control - 32, pp. 920-922.

[4]. Chen, Chi-Tsong, Linear System Theory and Design., (1984), Holt, Rinehart and Winston, NY.

[5]. Shrikhande, V. L., Mital, D. P. and Ray, L. H. (1980). On Minimal Canonical Realization from Input-Output Data Sequences, IEEE Trans. Aut. Control - 25, pp 309-312.

[6]. Luenberger, D. G. (1967). Canonical forms for Linear Multivariable Systems, IEEE Trans. Aut. Control-12, pp 290 - 293.

[7]. Sinha, N. K. and Kuszta, B., Modelling and Identification of Dynamic Systems (1983), Van Nostrand Reinhold.

21st Annual Pittsburgh
Conference on Modeling and
Simulation, May 3,4, 1990
Univ. of Pittsburgh

A Computational Procedure For Stabilizing Lightly Damped Systems

Bhaskar. M. Gorti , S. Bingulac , H. F. VanLandingham

**Bradley Department of Electrical Engineering
Virginia Polytechnic Institute and State University
Blacksburg , VA , 24061**

This work is supported by the Office of Naval Research under Grant N-00014-89-J-3123.

Abstract :

This paper presents a simple computational procedure for stabilizing lightly damped systems by infinitesimal shifting of the real parts of the eigenvalues of the system using a proportional output feedback regulator matrix. The shifting of the eigenvalues is achieved by continuous increments in the real parts of the system modes. The output feedback regulator matrix performing the shifting is obtained by solving the Lyapunov Algebraic Matrix equation. This approach facilitates stabilizing any particular or all modes of the system by shifting the corresponding eigenvalues as far as desired from the imaginary axes.

" A Computational Procedure for Stabilizing Lightly Damped Systems "

Consider a state space representation of a linear time-invariant multivariable dynamic system.

$$\dot{x} = \hat{A} x + \hat{B} u ; \quad y = \hat{C} x + \hat{D} u \quad (1)$$

where $x \in R^n$; $u \in R^m$; $y \in R^p$ are the state, input and output vectors respectively, while \hat{A} , \hat{B} , \hat{C} and \hat{D} are the matrices of compatible dimensions with \hat{B} and \hat{C} of maximum rank , i.e

$$\rho | \hat{B} | = m \quad \text{and} \quad \rho | \hat{C} | = p , \quad \text{respectively.} \quad (2)$$

Without loss of generality the following assumptions are made :

$$[1] . \quad m = p = \frac{n}{2} . \quad (3)$$

[2] . All eigenvalues λ_i , $i = 1, \dots, n$, of the matrix \hat{A} are distinct and have negative real parts.

[3] . All eigenvalues λ_i appear in complex-conjugate pairs, i.e $\lambda_{i+1} = \bar{\lambda}_i$, $i = 1, 3, \dots, n-1$, where $\lambda_i = -\sigma_i + j \omega_i$ and $\lambda_{i+1} = -\sigma_i - j \omega_i$.

[4] . The complex-conjugate pair $\{ \lambda_i , \lambda_{i+1} \}$, will be referred to as the j^{th} system mode m_j , where $j = \frac{(i+1)}{2}$. Thus, according to (3) , in \hat{A} there are p oscillatory modes,

$m_j = -\sigma_i \pm j \omega_i$; $j = 1, \dots, p$.

[5] . The system modes are arranged according to their magnitudes i.e :

$$|m_j| > |m_2| > \dots > |m_p| , \quad \text{where} \quad |m_j| = \sqrt{\sigma_j^2 + \omega_j^2} \triangleq \omega_{jn} .$$

With a sequence of similarity transformations (appendix 1) , a given system representation

$\{ \hat{A}, \hat{B}, \hat{C}, \hat{D} \}$ can be transformed into the representation $\{ A, B, C, D \}$ where ;

$$A = T \hat{A} T^{-1} ; \quad B = T \hat{B} ; \quad C = \hat{C} T^{-1} \quad \text{and} \quad D = \hat{D} . \quad (4)$$

The matrix A in (4) has the following structure :

$$A = \begin{bmatrix} O & : & I_m \\ \dots & : & \dots \\ -\Omega_O & & -D_O \end{bmatrix} \quad (5)$$

where O and I_m are $(m \times m)$ zero and identity matrices, respectively , while Ω_O and D_O are diagonal $(m \times n)$ matrices , i.e : $\Omega_O = \text{diag} \{ \omega_{jn} \}$ and $D_O = \text{diag} \{ 2 \sigma_j \}$ (6)

Infinitesimal Shifting of Real Parts of Eigenvalues λ_i

Using an $(m \times m)$ proportional output feedback regulator matrix K , i.e $u = -K y$ (7)

the closed loop system matrix A_c becomes : $A_c = A - B K C$ (8)

Partitioning B and C into $(p \times p)$ blocks B_1, B_2, C_1 and C_2 we get ;

$$B = \begin{bmatrix} B_1 \\ \dots \\ B_2 \end{bmatrix} \quad \text{and} \quad C = \begin{bmatrix} C_1 & : & C_2 \end{bmatrix} \quad (9)$$

The closed loop matrix A_c may be written as :

$$A_c = A - \begin{bmatrix} B_1 K C_1 & B_1 K C_2 \\ B_2 K C_1 & B_2 K C_2 \end{bmatrix} = \begin{bmatrix} -B_1 K C_1 & I_m - B_1 K C_2 \\ -\Omega_0 - B_1 K C_2 & -D_0 - B_2 K C_2 \end{bmatrix} \quad (10)$$

The output feedback regulator matrix K surely affects all m_j system modes, i.e all the system eigenvalues λ_j . The results of the above calculations are given in Appendix 2.

Consider the matrix A_c^* given by the following properties :

$$A_c^* = A + \begin{bmatrix} -\bar{D}_1 + E_1 & \vdots & E_3 \\ \dots & \dots & \dots \\ E_2 & \vdots & -\bar{D}_2 - E_1 \end{bmatrix} \quad (11)$$

where the matrices \bar{D}_1 , \bar{D}_2 and E_i , $i = 1, 2, 3$, have the following structures:

$$\bar{D}_1 = \text{diag} \{ 2\delta_j^1 \}, \quad \bar{D}_2 = \text{diag} \{ 2\delta_j^2 \}; \quad |\delta_j^i| \ll 1 \quad \text{and} \quad |E_i| \triangleq \epsilon_i; \quad \epsilon_i \ll 1. \quad (12)$$

Then, the modes m_j^c , i.e the eigenvalues λ_j^c , λ_{j+1}^c of the matrix A_c^* are such that

$$m_j^c \equiv m_j - \delta_j = -\sigma_j - \delta_j \pm j \omega_j; \quad \delta_j = \delta_j^1 + \delta_j^2 \quad (13)$$

The quantities δ_j could be interpreted as increments in the real parts of the modes m_j . Comparing (10) and (11), it can be concluded that in order to shift the modes m_j to the position m_j^c , the matrix K should satisfy the following equations:

$$B_1 K C_1 + B_2 K C_2 = \bar{D}_1 + \bar{D}_2$$

or

$$K C_1 (C_2)^{-1} + (B_1)^{-1} B_2 K = (B_1)^{-1} \bar{D} (C_2)^{-1}; \quad \text{where } \bar{D} = \bar{D}_1 + \bar{D}_2 \quad (14)$$

Note that the equation (14) is the Lyapunov Algebraic Matrix Equation, which could be solved for the unknown matrix K , provided that;

(a). The matrices B_1 and C_2 are non-singular and,

(b). All the eigenvalues λ_i^b and λ_j^c of the matrices $(B_1)^{-1} B_2$ and $C_1 (C_2)^{-1}$ satisfy the conditions $\lambda_i^b + \lambda_j^c \neq 0$, $\forall i$ and j .

In case of (2), conditions (a) and (b) are generically satisfied.

Defining $\bar{D} = \text{diag} \{ 2\bar{\Delta}_j \}$, $\bar{\Delta}_j \ll 1$, the equation (14) could be solved for the unknown K . From the equations (10) and (11) it could be concluded that the real parts of σ_j^c of the eigenvalues of the closed loop matrix A_c , (8), will satisfy the relation :

$$\sigma_j^c \approx \sigma_j + \bar{\Delta}_j \quad (15)$$

Stabilizing Lightly Damped Systems.

Assume that the system (1) is lightly damped , i.e its eigenvalues are relatively close to the imaginary axis. Also assume , that it is desired to move some of these eigenvalues farther left in the s-complex plane. This could be achieved by a sequence of infinitesimal shifting of the real parts as explained in the previous section.

The following algorithm is proposed:

[1] . Representation $\{ \hat{A}, \hat{B}, \hat{C}, \hat{D} \}$ of a lightly damped system is given .

Set the iteration counter $k = 1$.

The representation will be denoted as $\{ \hat{A}_1, \hat{B}_1, \hat{C}_1, \hat{D}_1 \}$.

Define a $(p \times p)$ zero matrix , K_t .

[2] . By the sequence of similarity transformations (Appendix 1) , the representation

$\{ \hat{A}_k, \hat{B}_k, \hat{C}_k, \hat{D}_k \}$ is transformed into $\{ A_k, B_k, C_k, D_k \}$, (4) ,

where A_k satisfies the property (5) .

[3] . Define a diagonal matrix $\tilde{D}_k = \text{diag} \{ 2\tilde{\Delta}_{jk} \}$, $0 \leq \tilde{\Delta}_{jk} \ll 1$. Non-zero $\tilde{\Delta}_j$ indicate which modes are desired to be infinitesimally shifted to the left in the k^{th} iteration.

[4] . Solve the Lypanov equation, i.e

$$K_k C_{1k} (C_{2k})^{-1} + (B_{1k})^{-1} B_{2k} K_k = (B_{1k})^{-1} \tilde{D}_k (C_{2k})^{-1} \quad (17)$$

Set $K_t + K_k = K_t$.

[5] . Build the closed loop system representation $\{ A_{ck}, B_k, C_k, D_k \}$, where

$$A_{ck} = A_k - B_k K_k C_k \quad (18)$$

Modes m_{jk}^c of the closed loop system in the k^{th} iteration are related to the modes m_{jk} of (16) by the relation : $m_{jk}^c \approx m_{jk} + \tilde{\Delta}_{jk}$.

[6] . Increment the iteration counter by 1, i.e set $k \approx k + 1$ and set

$$\hat{A}_k = A_{c(k-1)}, \hat{B}_k = B_{(k-1)}, \hat{C}_k = C_{(k-1)}, \hat{D}_k = D_{(k-1)}.$$

[7] . If the modes of the representation $\{ \hat{A}, \hat{B}, \hat{C}, \hat{D} \}$ are sufficiently shifted stop, ELSE

Go to step [2] .

** Appendix 1 and 2 , an Illustrative example, the Flowchart of the algorithm suggested and references will be included in the final version of the paper.

Estimation of NonGaussian Signals Using a Modified Gaussian Sum Adaptive Filter

Mauro J. Caputi, Richard L. Moose, William T. Baumann
Virginia Polytechnic Institute & State University
The Bradley Department of Electrical Engineering
Blacksburg, Virginia 24061

Abstract

The Gaussian sum estimation method developed by Sorenson and Alspach [1] is investigated. An alternate development of the Gaussian sum density approximation is given and a modified adaptive estimation structure is proposed based on the adaptive Kalman filter scheme first presented by Magill [2], and extended by Moose and Wang [3]. A necessary condition for effective estimation is examined [5]. Several alternate configurations are proposed when this condition is not met. Simulation results illustrating the theory are analyzed. A suboptimal method of tuning the parameters of the adaptive structure is suggested.

Introduction

An estimation technique, applicable to both linear systems with nonGaussian inputs and nonlinear systems with Gaussian inputs, has been developed by Sorenson and Alspach [1]. In the case of linear systems, the noise processes associated with the plant and measurement are assumed known and nonGaussian. The probability density function of each noise process is approximated by a Gaussian sum; that is, a weighted sum of Gaussian density functions. The Gaussian sum approximation is written as

$$p_{GS}(x) = \sum_{i=1}^M \alpha_i N[\mu_i, \sigma_i^2] \quad (1)$$

where

$$\sum_{i=1}^M \alpha_i = 1; \quad \alpha_i \geq 0 \text{ for } i = 1, 2, \dots, M \quad (2)$$

and

$$N[\mu_i, \sigma_i^2] = \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{1}{2}\left(\frac{x-\mu_i}{\sigma_i}\right)^2} \quad (3)$$

For sufficiently large M , any density function can be closely approximated by a Gaussian sum. As long as condition (2) holds, the Gaussian sum is always a valid density function.

Alternate Gaussian Sum Development

In developing their estimator, Sorenson and Alspach begin by directly representing the plant and measurement noise processes by Gaussian sums. The conditional density $p(x_k|Z_k)$ of the current state vector x_k given the current measurement sequence $Z_k = \{z_1, z_2, \dots, z_k\}$ is then updated using Bayes' rule.

A major problem with their development is that the number of terms in the Gaussian sum increases at each iteration of their estimator. This problem of *growing memory* can be avoided by the following alternate development of the Gaussian sum density approximation. The nonGaussian noise process or input signal is modeled as the sum of two statistically independent random processes

$$u = b + n \quad (4)$$

The first term, b , is a semi-Markov process with state transitions governed by the transition probability matrix of a conventional Markov process. However, the amount of time spent in state i before the next transition to state j is a random variable [4]. Its values lie within a *fixed set* of discrete values, characterized by a delta probability density function

$$p(b) = \sum_{i=1}^M P_i \delta(b - b_i) \quad (5)$$

with

$$\sum_{i=1}^M P_i = 1; \quad P_i \geq 0 \text{ for } i = 1, 2, \dots, M \quad (6)$$

This process can be thought of as a randomly-switching bias, each bias value b_i having probability P_i . The second term, n , is a zero mean Gaussian process with variance σ_i^2 . The variance switches at the same time the bias switches. This switching variance can be thought of as changing the power of the Gaussian process.

With both densities known, and using the convolution relationship between u , n , and b , the density function of u is given by

$$p_{GS}(u) = \sum_{i=1}^M P_i N[b_i, \sigma_i^2] \quad (7)$$

Thus, the nonGaussian density function of u can be modeled as a Gaussian sum. The

weight of each Gaussian term is the probability of the i^{th} biasing term. By allowing only a fixed number of bias values, the number of terms in the Gaussian sum at each iteration of the estimator is fixed, thereby eliminating the growing memory problem.

Selecting the parameters P_i , b_i , and σ_i in (7) to obtain the "best" approximation p_{GS} to some actual nonGaussian density function p is accomplished by means of minimizing the L^k norm

$$\|p - p_{GS}\|^k = \int_{-\infty}^{\infty} \left| p(u) - \sum_{i=1}^M P_i N[b_i, \sigma_i^2] \right|^k du \quad (8)$$

This curve fitting exercise can be done off-line using several values of M until a suitable trade-off between minimum norm and minimum M is obtained.

Development of a Modified Gaussian Sum Adaptive Filter

A modified Gaussian sum adaptive filter is now developed for a general linear system with a nonGaussian input signal and Gaussian plant and measurement noise processes. The system is modeled in standard discrete-time state-space form as

$$x_{k+1} = \Phi x_k + \Gamma u_k + \Psi w_k \quad (9)$$

$$z_k = H x_k + D u_k + v_k \quad (10)$$

$$u_k = b_k + n_k \quad (11)$$

where

x is the state vector

z is the measurement vector

w is a zero mean white Gaussian plant noise process with covariance Q

v is a zero mean white Gaussian measurement noise process with covariance R , independent of w

Φ, Γ, Ψ, H, D are the respective constant transition matrices

u is the vector Gaussian sum signal model (4) of the actual nonGaussian input signal, comprised of semi-Markov bias vector b_k , and zero mean white Gaussian noise n_k with covariance S_k

The optimal estimate of the state vector is found from the conditional mean as

$$\hat{x}_{k+1} = E[x_{k+1} | Z_{k+1}] = \int_{-\infty}^{\infty} x_{k+1} p(x_{k+1} | Z_{k+1}) dx_{k+1} \quad (12)$$

where Z_{k+1} is the measurement sequence $\{z_1, z_2, \dots, z_{k+1}\}$. Using Bayes' theorem, the conditional density function of (12) can be written as

$$p(x_{k+1} | Z_{k+1}) = \frac{p(x_{k+1}, Z_{k+1})}{p(Z_{k+1})} \quad (13)$$

The bias vector b_k and covariance matrix S_k are explicitly brought into (13) by considering the joint density $p(x_{k+1}, Z_{k+1})$ to be a marginal density found from $p(x_{k+1}, Z_{k+1}, b_k, S_k)$ by summing over the b_k and S_k terms

$$p(x_{k+1} | Z_{k+1}) = \frac{\sum_{i=1}^M p(x_{k+1}, Z_{k+1}, b_i, S_i)}{p(Z_{k+1})} \quad (14)$$

After some additional algebraic manipulation, the optimal estimate of (12) becomes

$$\hat{x}_{k+1} = \sum_{i=1}^M \hat{x}_{k+1}^i p(b_i, S_i | Z_{k+1}) \quad (15)$$

where \hat{x}_{k+1}^i is the conditional mean estimate of x_{k+1} given that $b_k = b_i$ and $S_k = S_i$, denoted by

$$\hat{x}_{k+1}^i = \int_{-\infty}^{\infty} x_{k+1} p(x_{k+1} | Z_{k+1}, b_i, S_i) dx_{k+1} \quad (16)$$

and the weighting factor $p(b_i, S_i | Z_{k+1})$ is the probability that $b_k = b_i$ and $S_k = S_i$ given the current measurement sequence. The overall estimate of (15) is then a weighted sum of individual estimates, each based on a particular set of parameters b_i and S_i .

Each estimate (16) is found by a modified Kalman filter. Inserting (11) into (9) and (10) produces

$$x_{k+1} = \Phi x_k + \Gamma b_k + w'_k \quad (17)$$

$$z_k = H x_k + D b_k + v'_k \quad (18)$$

where

$$w'_k = \Gamma n_k + \Psi w_k \quad (19)$$

$$v'_k = D n_k + v_k \quad (20)$$

Letting w_k , v_k , and n_k be statistically independent zero mean white Gaussian processes, and recognizing that $E[w'_{k-1} v'_k] = 0$, the Kalman filter equation for (15) is

$$\hat{x}_{k+1}^i = \Phi \hat{x}_k^i + \Gamma b_i + K_{k+1}^i [z_{k+1} - D b_i - H(\Phi \hat{x}_k^i + \Gamma b_i)] \quad (21)$$

with covariance and gain equations

$$M_{k+1}^i = \Phi P_k^i \Phi^T + \Gamma S_i \Gamma^T + \Psi Q \Psi^T \quad (22)$$

$$K_{k+1}^i = M_{k+1}^i H^T (H M_{k+1}^i H^T + D S_i D^T + R)^{-1} \quad (23)$$

$$P_{k+1}^i = (I - K_{k+1}^i H) M_{k+1}^i \quad (24)$$

The structure of the overall adaptive filter is a bank of Kalman filters operating in parallel, with each individual estimate multiplied by its own weighting term probability. This is the basic structure as outlined in Magill [2], with some modifications as will be shown. The filter based on the parameter set that most closely matches the actual parameters of the modeled system will have a weighting term that tends closer to one, while the weights of the mismatched filters will tend towards zero.

Using Bayes' rule and writing Z_{k+1} as $\{Z_k, z_{k+1}\}$, the weighting term becomes

$$p(b_i, S_i | Z_{k+1}) = \frac{p(z_{k+1} | b_i, S_i, Z_k) p(b_i, S_i | Z_k)}{p(z_{k+1} | Z_k)} \quad (25)$$

The denominator term is independent of i and is a scale factor that ensures that the sum of the weights (25) at each iteration is equal to one. The second term of the numerator can be rewritten to explicitly include the sample time for the bias and covariance terms

$$p(b_i, S_i | Z_k) = p(b_{k+1} = b_i, S_{k+1} = S_i | Z_k) \quad (26)$$

Using Bayes' rule and the definition of marginal densities, (26) can be written as

$$p(b_i, S_i | Z_k) = \sum_{j=1}^M p(b_{k+1} = b_i, S_{k+1} = S_i | b_k = b_j, S_k = S_j) p(b_k = b_j, S_k = S_j | Z_k) \quad (27)$$

The following notation will be used to express the two terms of the summation (27)

$$\Theta_{i,j} = p(b_{k+1} = b_i, S_{k+1} = S_i | b_k = b_j, S_k = S_j) \quad (28)$$

$$w_k^j = p(b_k = b_j, S_k = S_j | Z_k) \quad (29)$$

The density function of (28) is a Markov transitional probability matrix [4]; that is, $\Theta_{i,j}$ is the conditional probability that $b = b_i$ and $S = S_i$ at time $k+1$, given that $b = b_j$ and $S = S_j$ at time k . The density function of (29) is of the same form as (25) and is just the previous weighting term at the previous value of time k .

Necessary Condition for Effective Estimation

The first term of the numerator of (25) is of prime importance in determining whether or not the adaptive filter will work properly for a given system. It can be modeled as a Gaussian density if the bias and covariance terms switch slowly compared to the sample interval k . This assumption will be made here and has been verified by extensive simulation and analysis by Moose and Wang [3]. This conditional density is

$$p(z_{k+1} | b_i, S_i, Z_k) = N[H(\Phi \hat{x}_k^i + \Gamma b_i) + D b_i, H M_{k+1}^i H^T + D S_i D^T + R] \quad (30)$$

The i^{th} measurement residual from (21) is

$$r_{k+1}^i = z_{k+1} - D b_i - H(\Phi \hat{x}_k^i + \Gamma b_i) \quad (31)$$

The mean and covariance of (30) are readily available from the residual (31) and the Kalman gain (23). Essentially, the measurement residual of the matched filter will have the smallest expected value (typically, zero mean), while the residuals of all the mismatched filters will be biased. Under the Gaussian assumption, the probability of the matched filter will be the largest among all the filters.

For some systems, however, the statistical steady-state value (that is, the steady-state value of the expected value) of the measurement residual of *each* filter goes to zero. None of the residuals are biased. Therefore, the adaptive system cannot detect which filter has the correct parameter set and the weights all tend to the same value.

The statistical steady-state (SSS) value of the measurement residual is a function of the dc gain of the system transfer function [5]. If the dc gain of the system is zero, then the SSS value of the residual is zero. This condition provides a simple test as to whether or not this adaptive system will work properly for a particular system.

Alternate Estimation Procedures

A possible solution to this problem is to first determine if an alternate measurement provides a nonzero dc gain. For example, if in the zero dc gain system acceleration measurements are taken, changing to velocity measurements may provide a nonzero dc gain. If the actual sensors producing these measurements cannot be changed, integrating the acceleration measurement data to produce approximate-velocity measurement data may allow the adaptive system to work.

To illustrate this point, consider a general modal structure containing modes which are closely spaced in frequency and heavily damped. The input to the system is a nonGaussian stochastic FM signal, not directly measurable by the system. The model of this system follows (9-11). Checking the dc gain of this system using an acceleration measurement (with no noise)

$$G(\omega = 0) = G_0 = H(I - \sigma^{-1} \Gamma + D) \quad (32)$$

gives us zero dc gain. Since the SSS value of the measurement residual is a function of this dc gain

$$r_{ss}^i = [I - H\Phi(I - (I + K_{ss}H)\Phi)^{-1}K_{ss}]G_0(u_k - b_i) \quad (33)$$

the residuals for all filters go to zero when the system reaches SSS. Therefore, the residual density function $p(z_{k+1} | b_i, S_i, Z_k)$ cannot discern which filter is the matched filter and all the weighting terms (25) become equal.

Changing this modal system to output velocity measurement data produces a transfer function with a nonzero dc gain. Typically, changing the actual system cannot be done in practice. However, given a large signal to noise ratio between the input signal and measurement noise, a good approximation of the velocity measurement data can be generated by integrating the acceleration measurement data from the original system using a first or second order discrete time integrator. Redesigning the residual density function to be based on the velocity measurement model, while actually using *approximate*-velocity measurement data, will cause the weighting terms to properly select the matched filter.

Simulation Analysis of Alternate Adaptive Filters

Several simulations were run to test the performance of the Gaussian sum filter under various conditions. One simulation implements the velocity measurement system with nonzero dc gain, and uses an adaptive filter design based on the velocity measurement model. This serves as a benchmark for comparison with the other simulations.

A second simulation implements the acceleration measurement system with zero dc gain. Two adaptive filters are used. One filter is designed based on the acceleration measurement model, and produces the individual state estimates from (21). The second filter is designed based on a velocity measurement model, but uses *approximate*-velocity measurement data generated from a discrete time integrator. This filter is used to produce the proper weighting terms (25). The overall performance is not as good as the filter of the first simulation. The second filter could also be used to generate the individual state estimates, but the performance is no better than the first filter since the same acceleration measurement data is used to generate the *approximate*-velocity measurement data.

A third simulation implements a dual measurement system, where the measurement is comprised of two models: the acceleration measurement model, and the velocity measurement model. The adaptive filter is designed based on the dual measurement model and uses acceleration measurement data plus *approximate*-velocity measurement data. This filter can be tuned to perform better than the filter of the second simulation, but not as well as the filter of the first simulation.

Determination of Approximate Parameter Values

As mentioned previously, the system cannot directly measure the input signal. Referring to the stochastic FM input, the bias parameters b_i used to model its density

as a Gaussian sum are based on the amplitude of the signal. For good performance, the adaptive filter must have some approximate knowledge of this amplitude.

One possible method of determining amplitude information [5] is to monitor and process the measurement residual of the zero-bias filter (the individual filter with a bias parameter value of zero) of the weighting term adaptive filter of the second simulation described previously. Setting $b_i|_{i=0} = 0$ in (31) gives

$$r_{k+1}^0 = z_{k+1} - H\Phi\hat{x}_k^0 \quad (34)$$

The velocity measurement model, given below, of the weighting term adaptive filter is of the same form as (10), except $D = 0$

$$z_{k+1} = Hx_{k+1} + v_{k+1} \quad (35)$$

Using (9), (35), and setting the error $e_k^0 = (x_k - \hat{x}_k^0)$, the zero-bias residual (34) becomes

$$r_{k+1}^0 = H\Phi e_k^0 + H\Gamma u_k + H\Psi w_k + v_{k+1} \quad (36)$$

Given that the input signal u_k is much larger than both w_k and v_{k+1} , and then premultiplying both sides of (36) by $(H\Gamma)^{-1}$, the zero-bias residual is processed as

$$\tilde{u}_{k+1} = (H\Gamma)^{-1}r_{k+1}^0 \simeq (H\Gamma)^{-1}H\Phi e_k^0 + u_k \quad (37)$$

If the expected value of the first term on the right hand side of (37) is close to zero, then the expected value of the processed zero-bias residual can be approximated by

$$E[\tilde{u}_{k+1}] \simeq u_k \quad (38)$$

To approximate this expected value, a smoothed version of \tilde{u}_{k+1} is found from a first order discrete time smoother

$$\tilde{u}_{k+1}^s = \alpha_s \tilde{u}_k^s + (1 - \alpha_s) \tilde{u}_k \quad (39)$$

The peak values and power of an n -sample window of (39) are calculated and used to periodically update the bias parameters in the remaining nonzero-bias filters in the adaptive filter bank. This could serve as a suboptimal method of tuning the bias parameters to their "optimal" values when the input signal amplitude is originally unknown, or if the signal makes large changes in amplitude at unknown times.

References

- [1] H. W. Sorenson and D. L. Alspach, "Recursive Bayesian Estimation Using Gaussian Sums," *Automatica*, vol. 7, pp. 465-479, 1971.
- [2] D. T. Magill, "Optimal Adaptive Estimation of Sampled Stochastic Processes," *IEEE Trans. Auto. Cont.*, vol. AC-10, no. 4, Oct. 1965.
- [3] R. L. Moose and P. P. Wang, "An Adaptive Estimator with Learning for a Plant Containing Semi-Markov Switching Parameters," *IEEE Trans. on Sys., Man, and Cyber.*, vol. SMC-3, no. 3, May 1973.
- [4] R. A. Howard, *Dynamic Probabilistic Systems, Volume II: Semi-Markov and Decision Processes*. New York: Wiley, 1971.
- [5] M. J. Caputi, "NonGaussian estimation using a modified Gaussian sum adaptive filter," Ph.D. dissertation in preparation, Virginia Polytech. Inst. & S. U., Blacksburg, Virginia.

Computational Simplification in ERA (Eigensystem Realization Algorithm)

Bhaskar M. Gorti, S. Bingulac, H. F. VanLandingham

Bradley Department of Electrical Engineering

Virginia Polytechnic Institute and State University

Blacksburg, VA , 24061

Abstract.

A computational simplification of the Ho-Kalman minimal realization procedure, recently referred to as the Eigensystem Realization Algorithm (ERA), is proposed. According to the simplified algorithm, instead of performing the singular value decomposition (SVD) of the Hankel matrix and multiplying with matrices containing left and right singular vectors, a minimal system state space representation may be obtained by simple selecting appropriate rows or columns from the Hankel matrix. The obtained state representation is in either Pseudo-Controllable or Pseudo-Observable form.

Introduction.

In their seminal paper [1], Ho and Kalman suggested a procedure for calculating a minimal state space representation $R = \{A, B, C\}$ of a MIMO system from given Markov Parameters Y_i , defined by $Y_i = CA^i B$; $i = 0, 1, \dots, N$. Since publication of that paper a number of papers and textbooks, [1]-[8], have referred to this minimal realization procedure and have suggested slightly modified versions. Relatively recently, this algorithm has been given considerable attention, particularly in the area of large flexible structures, where it has received a name ERA (Eigensystem Realization Algorithm), [9]-[12]. In all these papers different versions were suggested, but all of them basically follow the following procedure.

1: Define Markov Parameters, i.e. $(p \times m)$ matrices Y_i , of an n^{th} order MIMO system with m inputs and p outputs.

2: Build the $(rp \times rm)$ Hankel matrix H_r given by, $H_r = \begin{bmatrix} Y_0 & Y_1 & \dots & Y_{r-1} \\ \vdots & \vdots & \ddots & \vdots \\ Y_{r-1} & Y_{2r-2} & \dots & Y_{2r-1} \end{bmatrix}$ for an arbitrary integer r .

3: Increase the r sequentially, until the rank of H_r does not increase, i.e. when $\rho(H_r) = \rho(H_{r+1})$, then $n = \rho(H_r)$ represents the order of a minimal state space representation.

4: Perform the SVD of H_r , i.e. calculate matrices U , Σ_n and V satisfying

$$H_r = U \begin{bmatrix} \Sigma_n & \dots & O \\ \vdots & \ddots & \vdots \\ O & \dots & O \end{bmatrix} V^T \text{ where } \Sigma_n = \text{diag}\{\sigma_1, \dots, \sigma_n\}, \sigma_i > 0$$

5: The matrices A , B , C in a minimal realization are given by $A = (\Sigma_n)^{-1/2} U_1^T \tilde{H}_r V_1 (\Sigma_n)^{-1/2}$,

$$B = (\Sigma_n)^{1/2} V_{11}$$

$$\text{where } \tilde{H}_r = \begin{bmatrix} Y_1 & \dots & Y_r \\ \vdots & \ddots & \vdots \\ Y_r & \dots & Y_{2r} \end{bmatrix} \text{ and } U_1, V_1, U_{11} \text{ and } V_{11} \text{ are } (rp \times n), (rm \times n), (n \times m) \text{ and } (p \times n) \text{ matrices respectively.}$$

$$C = U_{11} (\Sigma_n)^{1/2}$$

U_1 contains the first n columns from U and V_1 contains the first n columns from V . Similarly, U_{11} contains the first p rows from U_1 and V_{11} contains the first m columns from $(V_1)^T$.

Simplified ERA.

The computational simplification in ERA suggested in this paper is based on the use of pseudo controllability/observability indices, introduced recently. Consequently, as explained in [13]-[16], there are more equivalent state space representations in pseudo controllable/observable forms corresponding to the given Markov parameters, Y_i . Thus, after calculating the rank of H_r (steps 1,2,3 of the above algorithm), the only calculations that should be done, in the case of representing the system in a pseudo controllable form are : (i.) Select n appropriate columns from H_r with locations ν_1^c, \dots, ν_n^c ; $\nu_i^c < \nu_{i+1}^c$, into the $(rp \times n)$ matrix H_1 and select n columns from H_r with locations $m + \nu_1^c, \dots, m + \nu_n^c$, into the $(rp \times n)$ matrix H_2 . (ii.) The system matrix A_c in a pseudo controllable form is given by the least square solution of

$$H_1 A_c = H_2 \text{ or } A_c = (H_1^T H_1)^{-1} H_1^T H_2 \quad (1)$$

The input matrix B_c is fixed and is always given by $B_c = [I_m \ O]^T$, while the output matrix C_c contains the first p rows from H_1 . Similarly, in order to obtain a state space representation in a pseudoobservable form, the procedure is dual to the procedure for $\{A_c, B_c, C_c\}$. Obviously, the calculation required by (1), is much simpler than those required in the steps 4-5 of the original ERA algorithm. Particularly, when the system order is known or assumed calculation of SVD is not required. The n location numbers ν_1^c, \dots, ν_n^c , defining the columns of H_r to be selected in matrices H_1 and H_2 are uniquely related to the assumed set of admissible pseudo controllability indices, respectively, [15].

This work is supported by the Office of Naval Research under Grant N00014-89-J-3123.

Example

Consider the example given in [17], pp 491-492:

Given Markov Parameters $Y_i, i = 0, \dots, 5$: $\begin{bmatrix} -1 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 4 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 2 & 7 \\ 1 & 2 \end{bmatrix}$

The order of the minimal realization and the number of inputs and outputs are $n = 4, m = 2$ and $p = 2$. The possible sets of pseudo controllable indices are $\{1,3\}, \{2,2\}$ and $\{3,1\}$. The Hankel matrix H and the locations $\{\nu_i^c\}$, corresponding to the sets of pseudo controllability indices are:

$$H = \begin{bmatrix} Y_0 & \dots & Y_2 \\ \vdots & & \vdots \\ Y_2 & \dots & Y_5 \end{bmatrix} \quad \begin{array}{l} \text{p.c.i. : } \{1, 3\} \quad \{2, 2\} \quad \{3, 1\} \\ \text{locations : } \{1, 2, 4, 6\} \quad \{1, 2, 3, 4\} \quad \{1, 2, 3, 5\} \end{array}$$

The matrices H_1 and H_2 are formed using the above location indices, as defined in the earlier section. It is verified that the matrix H_1 in the case of p.c.i $\{2,2\}$ is not of full rank, therefore, this set of p.c.i is not admissible. According to (1), the pseudo controllable forms $\{A_c, B_c, C_c\}$ corresponding to pseudo controllability indices $\{1,3\}$ and $\{3,1\}$ are:

$$\text{p.c.i } \{1,3\} : \quad A_c = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 2 & 0 & 0 & 2 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \quad B_c = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad C_c = \begin{bmatrix} -1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{p.c.i } \{3,1\} : \quad A_c = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 2 & 0 & 2 \\ 1 & -1 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad B_c = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad C_c = \begin{bmatrix} -1 & 1 & 0 & 1 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

References

- [1] Ho, B. L. and Kalman, R. E., "Effective Construction of Linear State-Variable Models from Input/Output Data," *Proceedings of the 3rd Annual Allerton Conference on Circuit and System Theory*, 1965 pp. 449-459, also *Regelungstechnik*, vol. 14, 1966, pp. 545-548.
- [2] Chen, Chi-Tsong, "Linear System Theory and Design," 1984, Holt, Rinehart and Winston, N.Y.
- [3] Brogan, R. V., "Modern Control Theory," 1974, Quantum Publishers, Inc, N.Y.
- [4] Ackermann, J., "Sampled - Data Control Systems", Springer-Verlag, 1985.
- [5] Sinha, N. K. and Kustza, B., "Modeling and Identification of Dynamic Systems," 1984, Van Nostrand Reinhold.
- [6] Gopinath, B., "On the Identification of Linear Time-Invariant Systems from Input/Output Data," *Bell System Tech. J.*, vol. 48, 1969, pp. 1101 - 1113.
- [7] Budin, M. A., "Minimal Realization of Discrete Linear Systems from Input/Output Observation" *IEEE Trans. Automat. Contr.*, vol. AC-13, pp. 395-401, 1971.
- [8] Silverman, L. M., "Realization of Linear Dynamical Systems", *IEEE Trans. Automat. Contr.*, vol. ac-16, No. 6, 1971, pp. 554-567.
- [9] Denman, E. et al., "Identification of Large Space Structures on Orbit", *Final Report*, AFRPL TR-86-054, Sept 1986.
- [10] Juang, J. N. and Pappa, R. S., "An Eigensystem Realization Algorithm for Model Parameter Identification and Model Reduction," *Journal of Guidance, Control and Dynamics*, vol. 8, No. 5, Sept-Oct 1985, pp. 620-627.
- [11] Pappa, R. S. and Juang, J. N., "Galileo Spacecraft Modal Identification Using an Eigensystem Realization Algorithm," *The Journal of the Astronautical Sciences*, vol. 33, no. 1, Jan-March 1985, pp. 15-33.
- [12] *Proceedings of the NASA/JPL Workshop on Identification and Control of Flexible Space Structures*, San Diego, CA, June 4-6, 1984.
- [13] Gevers, M. and Wertz, V., "Uniquely Identifiable State-space and ARMA parametrization for Multi-variable Linear Systems," *Automatica*, 20, 1982, pp. 333-347.
- [14] Bingulac, S. and Krtolica, R. V., "An Algorithm for Simultaneous order and Parameter Identification in Multi-variable Systems," *IFAC-IFORS Symposium*, Beijing, PRC, pp.27-31, 1988.
- [15] Bingulac, S. and Krtolica, R. V., "On Admissibility of Pseudoobservability and Pseudocontrollability," *IEEE Trans. Automat. Contr.*, 1987, vol. AC-32, pp. 920-922.
- [16] Gorti, B. M., Bingulac, S. and VanLandingham, H. F., "Deterministic Identification of Linear Multi-variable Systems," *Proceedings of the 22nd Southeastern Symposium on System Theory*, March 11-13, 1990, pp. 1 - 5.
- [17] Kailath, T., "Linear Systems," 1980, Prentice - Hall, Inc.